



卓越工程师培养计划

▪ EDA ▪

<http://www.phei.com.cn>

张 涵 任秀华 王全景 张 超 编著



# 基于PROTEUS

## 的电路及单片机 设计与仿真

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书详细介绍 PROTEUS 软件在电路及单片机设计中的具体应用, 包括基础篇、结合篇和职场篇三部分。基础篇讲述电工学、电子电路、单片机原理及 PROTEUS 软件的编辑环境、电路设计与仿真基础等内容。结合篇详细介绍 PROTEUS 软件在模拟电路、数字电路中的设计与仿真, 以及如何使用 PROTEUS 实现单片机基本功能的设计与仿真。职场篇从项目开发的角度出发, 详尽介绍如何利用 PROTEUS 软件进行项目开发的设计与仿真。

本书适合从事电子电路设计的工程技术人员阅读, 也可作为高等院校相关专业的教学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

## 图书在版编目 (CIP) 数据

基于 PROTEUS 的电路及单片机设计与仿真 / 张涵等编著. —北京: 电子工业出版社, 2012. 7

(卓越工程师培养计划)

ISBN 978 - 7 - 121 - 17477 - 3

I. ①基… II. ①张… III. ①单片微型计算机 - 系统仿真 - 应用软件 IV. ①TP368. 1

中国版本图书馆 CIP 数据核字 (2012) 第 139965 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 徐 萍

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787 × 1092 1/16 印张: 16 字数: 410 千字

印 次: 2012 年 7 月第 1 次印刷

印 数: 4 000 册 定价: 39.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 前 言

PROTEUS 嵌入式系统仿真与开发平台是由英国 Labcenter 公司开发的,是目前世界上最先进、最完整的嵌入式系统设计、仿真平台。它是一款可视化的、支持多种微处理器且支持多种微处理器开发环境联合调试的软、硬件仿真软件。PROTEUS 除具有电子电路原理图设计与仿真、PCB 自动或人工布线等功能外,还具有微控制系统与外设的混合电路系统协同仿真的功能。

本书对 PROTEUS 软件功能进行详细介绍,与一般的软件教程的明显区别在于,在简单介绍电子电路及单片机相关专业基础知识的基础上,结合专业的应用,以实例的形式介绍 PROTEUS 的相关功能;在实例中学会菜单和工具的应用。

本书共分三大部分:基础篇、结合篇和职场篇。基础篇讲述电工学、电子电路、单片机原理及 PROTEUS 软件的编辑环境、电路设计与仿真基础等内容。结合篇详细介绍 PROTEUS 软件在模拟电路、数字电路中的设计与仿真,以及如何使用 PROTEUS 实现单片机基本功能的设计与仿真。职场篇从项目开发的角度出发,详尽介绍如何利用 PROTEUS 软件进行项目的设计与仿真。

本书面向的是电子电路工业领域的学生和工程技术人员,包括一些 EDA 设计软件的初级用户、具有一定基础知识初入职场电子线路设计爱好者,以及高等院校相关专业的大中专学生、研究生。

本书共分 13 章。第 1 章介绍电工学基础知识,第 2 章介绍电子电路设计基础,第 3 章介绍 51 单片机的结构与原理,第 4 章介绍 PROTEUS 电路设计与仿真基础,前 4 章为基础篇;第 5 章至第 9 章为结合篇,主要介绍 PROTEUS ISIS 电路仿真、基于 PROTEUS ISIS 的电工基础的电路仿真、模拟电子电路的仿真与设计、基于 PROTEUS ISIS 的数字电路的仿真与分析及基于 PROTEUS ISIS 的单片机仿真;最后 4 章为职场篇,第 10 章为低频信号发生器的设计与仿真实例,第 11 章为基于 PCF8593 设计一个实时时钟系统,第 12 章是一个多单片机通信的应用设计与仿真,第 13 章介绍 PID 温度控制系统的设计及仿真实例。本书从实用角度出发,采用理论讲解与实例相结合的讲述方法,简明清晰、重点突出。在叙述上力求深入浅出、通俗易懂。相信会为读者的学习和工作带来一定的帮助。

本书由山东建筑大学的张涵、任秀华、王全景和张超编著。参加本书编写的还有管殿柱、李文秋、宋一兵、王献红、谈世哲、张轩、付本国、赵景波、赵景伟等。

由于编著者水平有限,书中错误疏漏在所难免,恳请广大读者批评指正。

编 著 者





# 目 录

## 基 础 篇

第 1 章 电工学基础	2
1.1 电路的基本定律	2
1.2 电路的分析方法	4
1.3 交流电路	12
1.4 思考与练习	16
第 2 章 电子电路设计基础	18
2.1 信号放大	18
2.2 信号滤波	24
2.3 组合逻辑电路	29
2.4 触发器	33
2.5 思考与练习	40
第 3 章 51 单片机的结构与原理	43
3.1 51 单片机的硬件结构	43
3.2 51 单片机的中断系统	53
3.3 51 单片机的串行接口及串行通信技术	57
3.4 51 单片机的定时/计数器	64
3.5 思考与练习	69
第 4 章 PROTEUS 电路设计与仿真基础	70
4.1 PROTEUS 软件简介	70
4.2 PROTEUS 软件的基本操作	74
4.3 PROTEUS ISIS 参数设置	76
4.4 基于 PROTEUS 的电路设计	81
4.5 元件创建	85
4.6 PROTEUS ISIS 原理图输入	98
4.7 虚拟仿真工具	100
4.8 库元件认识	104
4.9 思考与练习	112

## 结 合 篇

第 5 章	PROTEUS ISIS 电路仿真 .....	114
5.1	交互式仿真 .....	114
5.2	基于图表的仿真 .....	122
5.3	思考与练习 .....	128
第 6 章	基于 PROTEUS ISIS 的电工基础的电路仿真 .....	129
6.1	基于 PROTEUS ISIS 的直流电路欧姆定律的设计与仿真 .....	129
6.2	基于 PROTEUS ISIS 的基尔霍夫定律的设计与仿真 .....	130
6.3	基于 PROTEUS ISIS 的戴维南定理的设计与仿真 .....	133
6.4	基于 PROTEUS ISIS 的诺顿定理的设计与仿真 .....	134
6.5	基于 PROTEUS ISIS 的三相交流电路的设计与仿真 .....	135
6.6	思考与练习 .....	136
第 7 章	模拟电子电路的仿真与设计 .....	137
7.1	模拟电路常用器件与仪器 .....	137
7.2	单管共射放大器及负反馈 .....	142
7.3	射极跟随器 .....	148
7.4	差动放大器 .....	151
7.5	思考与练习 .....	156
第 8 章	基于 PROTEUS ISIS 的数字电路的仿真与分析 .....	157
8.1	基于 PROTEUS ISIS 的分立元件门电路的设计与分析 .....	157
8.2	基于 PROTEUS ISIS 的组合逻辑电路的设计与仿真 .....	161
8.3	基于 PROTEUS ISIS 的编码器电路的设计与仿真 .....	165
8.4	基于 PROTEUS ISIS 的译码器电路的设计与仿真 .....	167
8.5	基于 PROTEUS ISIS 的 D 触发器电路的设计与仿真 .....	168
8.6	思考与练习 .....	170
第 9 章	基于 PROTEUS ISIS 的单片机仿真 .....	171
9.1	单片机程序的建立与编译 .....	171
9.2	单片机电路的仿真与调试 .....	173
9.3	PROTEUS ISIS 与 Keil C 的联合调试 .....	175
9.4	I/O 口的输入/输出应用 .....	178
9.5	按键识别 .....	180
9.6	数码管显示 .....	184
9.7	外部中断实验 .....	186
9.8	定时/计数器应用 .....	190

9.9 思考与练习 .....	196
-----------------	-----

## 职 场 篇

第 10 章 低频信号发生器的设计与仿真 .....	199
10.1 系统功能 .....	199
10.2 设计方案 .....	199
10.3 系统硬件仿真电路 .....	201
10.4 程序设计 .....	201
10.5 软件调试与运行结果 .....	202
10.6 源程序清单 .....	203
10.7 思考与练习 .....	206
第 11 章 基于 PCF8593 的实时时钟的设计与仿真 .....	207
11.1 系统功能 .....	207
11.2 设计方案 .....	208
11.3 系统硬件仿真电路 .....	208
11.4 程序设计 .....	209
11.5 软件调试与运行结果 .....	210
11.6 PCF8593 精度的调整 .....	210
11.7 源程序清单 .....	211
11.8 思考与练习 .....	217
第 12 章 主从式多单片机通信系统的设计与仿真 .....	218
12.1 设计方案 .....	218
12.2 系统硬件仿真电路 .....	218
12.3 程序设计 .....	219
12.4 软件调试与运行结果 .....	220
12.5 源程序清单 .....	221
12.6 思考与练习 .....	224
第 13 章 基于 PID 控制算法的温度控制系统的设计与仿真 .....	225
13.1 设计方案 .....	225
13.2 系统硬件仿真电路 .....	226
13.3 程序设计 .....	230
13.4 软件调试与运行结果 .....	235
13.5 源程序清单 .....	236
13.6 思考与练习 .....	244
参考文献 .....	245

# 基 础 篇

# 第1章 电工学基础

目前,电工技术应用极为广泛,发展非常迅速,并且日益渗透到其他学科领域,促进其发展,在我国社会主义现代化建设中占有重要地位。通过对本章内容的学习,读者可获得电工技术必要的基本理论、基本知识和基本技能。本章将介绍电路的基本定律、电路的分析方法及交流电路的相关知识。




## 1.1 电路的基本定律

### 1. 理想电路元件

理想电路元件是电路模型中不可再分割的基本构造单元,具有精确的数学定义。理想电路元件也是一种科学的抽象,可以用来表征实际电路中的各种电磁性质。例如,“电阻元件”表征了电路中消耗电能的电磁特性;“电感元件”表征了电路中储存磁场能量的电磁特性;“电容”元件则表征了电路中储存电场能量的电磁特性。

实际电路中实体部件上发生的电磁现象往往是复杂的、多元的,如电阻器、电炉等设备,它们除了具有消耗电能的特性外,还有磁场和电场方面的特性,分析时若把它们的全部电磁特性都表征出来既有困难也不必要。本着突出主要矛盾、忽略次要因素的研究方法,电阻器和电炉等设备完全可以用理想的“电阻元件”来作为它们的数学模型。显然,理想电路元件是从实际电路器件中科学抽象出来的假想元件,可以看做是实际电路器件的一种“近似”。

理想电路元件简称为电路元件。虽然它们只能是实际电路器件的一种近似,但用它们及它们的组合可以相当精确地表征出实体电路器件的主要电磁特性。如工频条件下的电感线圈,其电路模型可以用一个“电阻元件”和一个“电感元件”的串联组合来表征;一个实际的直流电压源的电路模型则可以用一个“电阻元件”和一个“理想电压源”的串联组合来表征,等等。

 **注意:** 学习时,应理解各种理想电路元器件的严格定义,区分实际电路元器件与理想电路元件之间的联系和差别。本书中若无特殊说明,各理想电路元件均是指线性元件。

### 2. 欧姆定律

对任意一段电路,电流与该段电路两端的电压成正比,与该段电路中的电阻成反比。这一结论是在1827年由德国科学家欧姆提出的,因此称为欧姆定律。当电压与电流为关联参考方向时,欧姆定律可表示为:

$$I = \frac{U}{R} \quad (1-1)$$

上式仅适用于线性电路,它体现了线性电路元件上的电压、电流约束关系,表明了元件

特性只取决于元件本身，与其接入电路的方式无关这一规律。

### 3. 基尔霍夫定律

电路的基本定律除了欧姆定律，还有节点电流定律（KCL）和回路电压定律（KVL），KCL 和 KVL 都是德国科学家基尔霍夫提出的，因此也把 KCL 称为基尔霍夫第一定律，把 KVL 称为基尔霍夫第二定律。1847 年，基尔霍夫将物理学中“流体流动的连续性”和“能量守恒定律”用于电路之中，创建了节点电流定律（KCL），之后根据“电位的单值性原理”又创建了回路电压定律（KVL）。欧姆定律体现了电路元件上的电压、电流约束关系，与电路的连接方式无关；而基尔霍夫定律则反映了电路整体的规律，具有普遍性，不但适合于任何元件组成的电路，而且适合于任何变化的电压与电流，基氏两定律和欧姆定律被人们称为电路的三大基本定律。

基尔霍夫定律是集总电路的基本定律，包括电流定律和电压定律。它是分析一切集总参数电路的根本依据，一些重要的定理、有效的电路分析方法，都是以基尔霍夫定律为“源”推导、证明、归纳、总结后得出的。由于涉及元件的互连形式，故先介绍电路模型中的几个术语，然后再介绍基尔霍夫定律。

几个有关的常用电路术语如下。

- 支路：任意两个节点之间无分叉的分支电路称为支路，如图 1-1 中的  $baf$  支路、 $be$  支路、 $bcde$  支路。
- 节点：电路中，3 条或 3 条以上支路的交汇点称为节点，如图 1-1 中的  $b$  点、 $e$  点。
- 回路：电路中由若干条支路构成的任一闭合路径称为回路。如图 1-1 中的  $abefa$  回路、 $bcdeb$  回路、 $abcdefa$  回路。
- 网孔：不包围任何支路的单孔回路称为网孔。如图 1-1 中  $abefa$  回路和  $bcdeb$  回路都是网孔，而  $abcdefa$  回路不是网孔。网孔一定是回路，而回路不一定是网孔。

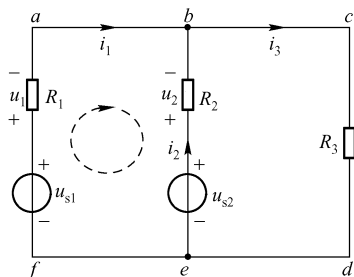


图 1-1 电路举例

**1) 基尔霍夫电流定律** 基尔霍夫电流定律（KCL）是用来反映电路中任意节点上各支路电流之间关系的。可表述为：对于任何电路中的任意节点，在任意时刻，流过该节点的电流之和恒等于零。其数学表达式为：

$$\sum i = 0 \quad (1-2)$$

如果选定电流流出节点为正，流入节点为负，如图 1-1 中的  $b$  节点，有  $-i_1 - i_2 + i_3 = 0$ 。将上式变换得： $i_1 + i_2 = i_3$ 。

所以，基尔霍夫电流定律还可以表述为：对于电路中的任意节点，在任意时刻，流入该节点的电流总和等于从该节点流出的电流总和。即

$$\sum i_i = \sum i_o \quad (1-3)$$

KCL 不仅适用于电路中的任一节点，也可推广应用于广义节点，即包围部分电路的任一闭合面。可以证明流入或流出一闭合面电流的代数和为零。

图 1-2 中，对于虚线所包围的闭合面，可以证明有如下关系： $I_a + I_b + I_c = 0$ 。

基尔霍夫电流定律是电路中连接到任一节点的各支路电流必须遵守的约束，而与各支路上的元件性质无关。这一定律对

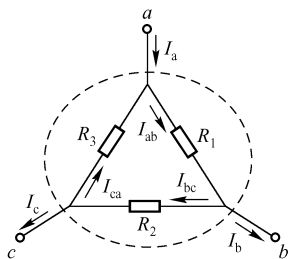


图 1-2 广义节点

于任何电路都普遍适用。

**2) 基尔霍夫电压定律** 基尔霍夫电压定律 (KVL) 是反映电路中各支路电压之间关系的定律。可表述为: 对于任何电路中的任一回路, 在任一时刻, 沿着一定的循行方向 (顺时针方向或逆时针方向) 绕行一周, 各段电压的代数和恒为零。其数学表达式为:

$$\sum u = 0 \quad (1-4)$$


如图 1-1 所示闭合回路中, 沿 *abefa* 顺序绕行一周, 则有:  $-u_{s1} + u_1 - u_2 + u_{s2} = 0$ 。式中,  $u_{s1}$  之前之所以加负号, 是因为按规定的循行方向, 由电源负极到正极, 属于电位升;  $u_2$  的参考方向与  $i_2$  相同, 与循行方向相反, 所以也是电位升;  $u_1$  和  $u_{s2}$  与循行方向相同, 是电位降。当然, 各电压本身还存在数值的正负问题, 这是需要注意的。

由于  $u_1 = R_1 i_1$  和  $u_2 = R_2 i_2$ , 代入上式, 有  $-u_{s1} + R_1 i_1 - R_2 i_2 + u_{s2} = 0$  或  $R_1 i_1 - R_2 i_2 = u_{s1} - u_{s2}$ 。

这时, 基尔霍夫电压定律可表述为: 对于电路中的任一回路, 在任一时刻, 沿着一定的循行方向 (顺时针方向或逆时针方向) 绕行一周, 电阻元件上电压降之和恒等于电源电压升之和。其表达式为:

$$\sum Ri = \sum u_s \quad (1-5)$$

按式 (1-5) 列回路电压平衡方程式, 当绕行方向与电流方向一致时, 该电阻上的电压取 “+”, 否则取 “-”; 当从电源负极循行到正极时, 该电源参数取 “+”, 否则取 “-”。

 **注意:** 应用 KVL 时, 首先要标出电路各部分电流、电压或电动势的参考方向。列电压方程时, 一般约定电阻的电流方向和电压方向一致。

KVL 不仅适用于闭合电路, 也可推广到开口电路。图 1-3 中, 有  $U = 2I + 4$ 。

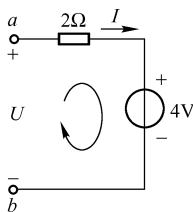


图 1-3 开口电路



## 1.2 电路的分析方法

电路的基本概念和基本定律, 是电路分析理论中的共同约定和共同语言。但是, 由于工程实际中应用电路的结构多种多样, 求解的对象也往往由于具体要求的不同而大相径庭, 所以, 只用电路的基本概念和基本定律来分析和计算较为复杂的电路, 显然是不够的。为此, 介绍几种常用电路的分析方法。

电路分析方法大致可分为两类: ①等效变换及其化简的方法, 如实际电源间的等效变换、含源单口网络的等效化简、电源转移法、T-p 变换; ②列解网络方程和定理的分析法, 也称为电路的基本分析法, 如支路电流法、支路电压法、基本割集分析法 (特别节点分析法)、基本回路分析法 (特别网孔分析法)、叠加定理、戴维南和诺顿定理。另外, 动态电

路分析中, 还要涉及动态电路的时域经典分析法。

### 1. 电路的等效变换

等效电路是电路分析中一个很重要的概念, 应用它通过等效变换, 可以把多元件组成的电路化简为只有少数几个元件组成的单回路或一对节点的电路, 甚至单元件电路。它是化繁为简、化难为易的钥匙。下面将介绍无源二端电阻串、并联网络的等效, 无源三端网络 T 形和  $\pi$  形的等效变换及简单有源二端网络的等效变换。

**1) 无源二端电阻串、并联网络的等效** 单个二端元件是二端网络最简单的形式。无论是二端元件还是二端网络均有用各自的端钮间电压和端钮上电流所表示的伏安关系。

(1)  $n$  个电阻串联所组成的二端网络  $N_1$ 。如图 1-4 所示, 根据 KVL, 其端钮上的伏安关系为:

$$u = (R_1 + R_2 + \cdots + R_n) i = Ri$$

故得等效二端网络  $N_2$ , 如图 1-5 所示, 其等效电阻为:

$$R = R_1 + R_2 + \cdots + R_n \quad (1-6)$$

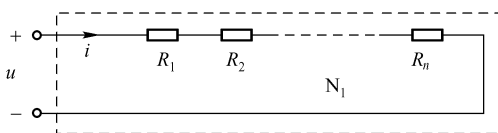


图 1-4  $n$  个电阻串联网路  $N_1$

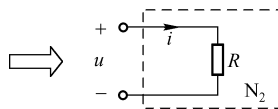


图 1-5 等效图

$N_1$  和  $N_2$  等效, 则外接同一电压  $u$ , 两者吸收相同的功率, 即

$$P = (R_1 + R_2 + \cdots + R_n) i^2 = Ri^2 \quad (1-7)$$

电阻串联存在着分压规律, 分压公式为:

$$u_k = \frac{R_k}{R} u \quad (1-8)$$

(2)  $n$  个电导并联所组成的二端网络。如图 1-6 所示, 根据 KCL, 其端钮上的伏安关系为:

$$i = (G_1 + G_2 + \cdots + G_n) u = Gu \quad (1-9)$$

故得等效二端网络  $N_2$ , 如图 1-7 所示, 其等效电导为:

$$G = G_1 + G_2 + \cdots + G_n \quad (1-10)$$

$N_1$  和  $N_2$  等效, 则外接同一电流  $i$ , 两者吸收相同的功率, 即

$$P = (G_1 + G_2 + \cdots + G_n) u^2 \quad (1-11)$$

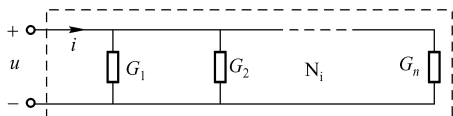


图 1-6  $n$  个电阻并联网络  $N_1$

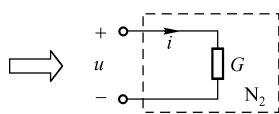


图 1-7 等效图

电导并联存在着分流规律, 分流公式为:

$$i_k = i \times \frac{G_k}{G} \quad (1-12)$$

无源二端电阻网络的结构很灵活, 解题时应注意以下几点:



- 研究电路结构是否对称；
- 让一电流从待求端口流进和流出，弄清连接关系及等电位点；
- 电位相等的节点重合或用短路线连通；
- 无电流的支路开路，阻值不计。

这样就能对复杂电阻网络进行正确的等效化简。

2) 无源三端网络 T 形和  $\pi$  形的等效变换 二端网络的等效原则可在三端网络推广，如图 1-8 所示为 T 形和  $\pi$  形网络。

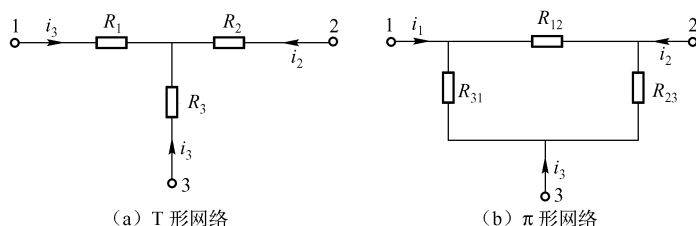


图 1-8 T 形和  $\pi$  形网络

根据 KVL 和 KCL，若两个三端网络对应端间的电压相等，对应端的电流相同，即伏安关系相同，则这两个三端网络等效。据此可推出其等效变换公式如下：

(1)  $\pi$  形变 T 形等效变换公式：

$$R_1 = \frac{R_{12}R_{31}}{R_{12} + R_{23} + R_{31}}; R_2 = \frac{R_{23}R_{12}}{R_{12} + R_{23} + R_{31}}; R_3 = \frac{R_{23}R_{31}}{R_{12} + R_{23} + R_{31}} \quad (1-13)$$

(2) T 形变  $\pi$  形等效变换公式：

$$R_{12} = \frac{R_{12} + R_{23} + R_{31}}{R_3}; R_{23} = \frac{R_{12} + R_{23} + R_{31}}{R_1}; R_{31} = \frac{R_{12} + R_{23} + R_{31}}{R_2} \quad (1-14)$$

当 3 个电阻相等时，互换公式为：

$$R_T = \frac{1}{3}R_\pi \quad (1-15)$$

3) 简单有源二端网络的等效变换 单个理想电源或实际电源所表示的含源支路是有源二端网络最简单的形式。一般来说，一切含源二端网络均可等效为电压源和电阻的串联组合电路或电流源和电阻的并联组合电路。下面就对含源二端网络的等效变换论述如下。

(1) 两种实际电源模型的等效变换。实际的电源总是有内阻的，它的端电压（电流）不可能为定值，因此常用一个理想电压源  $u_s$  和内阻  $R_s$  相串联来表征一个实际电压源，如图 1-9 所示。

常用一个理想电流源  $i_s$  和内阻  $R_s$  相并联的模型来表征一个实际电流源，如图 1-10 所示。

比较这两种实际电源的伏安关系，若  $R_s$  相等， $i_s = \frac{u_s}{R_s}$ ，则这两种电源外端钮上的伏安关系相同，即两者等效。具体等效变换如图 1-10 所示。

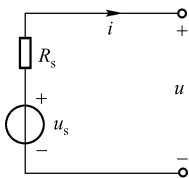


图 1-9 电压源模型

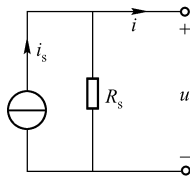


图 1-10 电流源模型

(2) 简单的含受控源的二端网络的等效化简。受控源的两个端钮和独立源一样,有自己的伏安关系,因此根据二端网络等效的定义,含受控源的二端网络,可像含独立源的一样等效化简。不过应注意以下两点:

- 受控源的电压和电流受电路中其他支路的电压或电流的控制,是控制量的函数,因此化简时应保留控制量支路。
- 等效变换是一种重要的电路分析方法,但只对具有一定结构形式的简单电路行之有效。对于较复杂的电路,必须有一些更普遍、更一般的分析手段。

## 2. 戴维南定理和诺顿定理

下面以欧姆定律和基尔霍夫定律为基础,寻求不同的电路分析方法。线性网络的分析方法有两种:一种是以 KCL 和 KVL 为基础的分析方法,如支路电流法、回路电流法、网孔电流法、节点电压法;另一种是电路定理,利用电路定理将复杂电路化简或将电路的局部用简单电路等效替代,以使电路的计算得到简化。这种方法有叠加定理、戴维南定理等。

**1) 支路电流法** 支路电流客观存在于电路之中,直接把它设为未知变量,然后应用 KCL 和 KVL 定律对电路列写方程式进行求解,即可求得各支路电流,再在各支路电流基础上按要求求取其他电路变量。这种解题方法称为支路电流法。

(1) 支路电流法的分析步骤。

- ① 以各支路电流为未知变量,选定标注其参考方向。
- ② 任意选定  $n-1$  个节点,按 KCL 列出  $n-1$  个独立的节点 KCL 方程。
- ③ 选定  $l=b-(n-1)$  个基本回路并规定标注其绕行方向,结合元件特性并按 KVL 列出  $b-(n-1)$  个独立的回路 KVL 方程。

④ 联立求解以上  $b$  个相互独立的电路方程,得到各支路电流。

⑤ 根据分析要求,以支路电流为基础求取其他电路变量。

(2) 应用注意事项。

- 支路电流法适用于分析含受控源的电路;
- 支路电流法要求每条支路的电压可用支路电流来表示,否则须另行处理。例如,在一条支路仅含电流源且不直接与任何电阻并联的情况下,可以增加一个未知变量(电流源两端电压),最后再补充一个电路方程(该支路电流等于电流源电流)。

(3) 支路电流法的特点。由支路电流法分析思想可知,为求取各支路电流,必须列出  $b$  个相互独立的电路方程。一个电路包含的支路数目越多,求取各支路电流所需的电路方程数目就越多,解方程组的难度也就越大。因此支路电流分析法宜于利用计算机求解。人工计算时,对于支路数不太多的电路,可以应用支路电流法,但为进一步减少分析所需的电路方程数量,可采用网孔电流法和节点电压法。

**2) 回路电流法** 当一个复杂电路的支路数较多时,需要列写较多个方程式,造成解题过程的烦琐和不易。观察图 1-11 所示电路,该电路虽然支路数较多,但网孔数却较少。针对上述类型电路,为了适当减少方程式的数目,引入回路电流法。

以图 1-11 所示电路为例,将它重画在图 1-12 中,由于此电路具有 4 个节点、6 条支路、7 个回路和 3 个网孔,因此利用支路电流法求解时,需列出 3 个 KCL 方程式和 3 个 KVL 方程式,而对 6 个方程式进行联立求解时,其过程的烦琐程度可想而知。

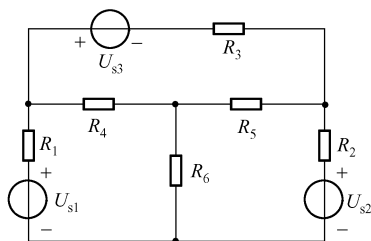


图 1-11 回路电路举例 1

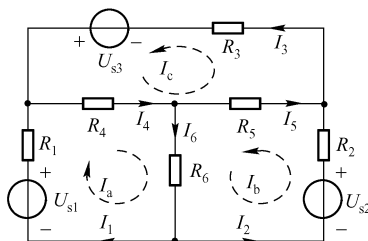


图 1-12 回路电流举例 2

如果假想在 3 个独立回路中（独立回路一般选取独立网孔），各自有一个绕回路环行的电流，把这些假想的绕回路流动的电流取名为回路电流，如图 1-12 电路中虚线箭头所示的  $I_a$ 、 $I_b$  和  $I_c$ 。由于回路电流在流入和流出节点时并不发生变化，因此它们自动满足 KCL 定律。这样，在求解电路时，KCL 方程式就被省略了，只需对 3 个独立回路列出相应的 KVL 方程式即可。

选定图 1-12 所示电路的 3 个网孔作为独立回路，列写 3 个 KVL 方程式如下：

$$\text{对回路 a: } (R_1 + R_4 + R_6)I_a + R_4I_c + R_6I_b = U_{s1}$$

$$\text{对回路 b: } (R_2 + R_5 + R_6)I_b - R_5I_c + R_6I_a = U_{s2}$$

$$\text{对回路 c: } (R_3 + R_4 + R_5)I_c - R_5I_b + R_4I_a = U_{s3}$$

3 个方程式的左边为电阻压降，其中第一项为本回路电流流经本回路中所有电阻时产生的电压降，括号内的所有电阻称为回路的自电阻；方程式左边的第二项和第三项，为相邻回路电流流经本回路公共支路上连接的电阻（即  $R_4$ 、 $R_5$  和  $R_6$ ）时产生的电压降，把这些公共支路上连接的电阻称为互电阻。换句话说，每一个互电阻上的电压降都是相邻两个回路电流在互电阻上产生的电压的叠加。

上述问题并不难理解，仔细观察电路中客观存在的支路电流  $I_1 \sim I_6$ ，找出它们与假想的回路电流之间的关系，可以看出：

$$\begin{aligned} I_1 &= I_a & I_4 &= I_a + I_c \\ I_2 &= I_b & I_5 &= I_c - I_b \\ I_3 &= I_c & I_6 &= I_a + I_b \end{aligned}$$

也就是说，实际上互电阻  $R_4$  上的电压降是  $I_4R_4$ ，对应回路电流产生的压降是  $I_aR_4 + I_cR_4$ ；互电阻  $R_5$  上的电压降是  $I_5R_5$ ，对应回路电流产生的压降是  $I_cR_5 - I_bR_5$ ；互电阻  $R_6$  上的电压降是  $I_6R_6$ ，对应回路电流产生的压降是  $I_aR_6 + I_bR_6$ 。即回路电流法中的 3 个 KVL 方程式，实质上与支路电流法中的 3 个 KVL 方程式完全等效，只不过用假想的、实际上并不存在的回路电流替代了客观存在的支路电流。方程式的右边，由于不涉及回路电流，因此与支路电流法中 KVL 方程式的右边完全相同。

对多支路少回路的平面电路而言，以回路电流为未知量，根据 KVL 列写回路电压方程，求解出回路电流，进而求出客观存在的支路电流、电压或功率等的解题方法，称为回路电流法。提出回路电流法的目的是为了对图 1-12 所示电路进行分析和计算时，减少方程式的数目，当一个电流的支路数与回路数相差不大时，采用回路电流法显然意义不大了。

归纳回路电流法求解电路的基本步骤如下。

① 选取独立回路（一般选择网孔作为独立回路），在回路中标示出假想回路电流的参考

方向,并把这一参考方向作为回路的绕行方向。

② 建立回路的 KVL 方程式。应注意自电阻压降恒为正值,公共支路上互电阻压降的正、负由相邻回路电流的方向来决定:相邻回路电流方向流经互电阻,与本回路电流方向一致时该部分压降取正,相反时取负。方程式右边电压升的正、负取值方法与支路电流法相同。

③ 求解联立方程式,得出假想的各回路电流。

④ 在电路图上标出客观存在的各支路电流的参考方向,按照它们与回路电流之间的关系,求出各条支路电流。

**3) 网孔电流法** 所谓网孔电流是一种为了简化电路分析而人为引入的电路变量,是一种假想的沿着网孔连续流动的电流(实际上不存在网孔电流而只有支路电流)。

(1) 网孔电流的特点。

- 电路中实际上不存在网孔电流,网孔电流是一种假想电流。
- 引入网孔电流旨在简化电路分析,减少分析所需的电路独立方程数。
- 引入网孔电流后,电路中各支路电流可以用网孔电流来表示。
- 一般指定网孔电流的参考方向与网孔的绕行方向一致。
- 电路中各网孔电流在任意节点自动满足 KCL。

(2) 网孔电流分析法的基本思想。对于一个含  $b$  条支路、 $n$  个节点的平面电路,从电路图中可知,其网孔数为  $l = b - (n - 1)$ ,显然  $l < b$ 。因此如果假想有一种电流沿网孔流动,以网孔电流为未知变量,根据 KVL 列出  $l = b - (n - 1)$  个网孔的 KVL 方程(由于网孔电流自动满足 KCL,故不必再列节点 KCL 方程),联立求得各网孔电流,最后根据网孔电流的特点求取支路电流及其他电路变量。

(3) 网孔电流法的分析步骤。

- ① 确定各网孔电流,指定其参考方向并以参考方向作为网孔的绕行方向。
  - ② 按 KVL 列写  $l = b - (n - 1)$  个网孔的 KVL 方程 [(共  $l = b - (n - 1)$  个且相互独立)]。
  - ③ 联立求解得到各网孔电流。
  - ④ 在所得网孔电流基础上,按分析要求求取其他电路变量。
- (4) 几种特殊情况。

- 电路中含有电流源与电阻的并联组合:将其等效变换成电压源与电阻的串联组合后列电路方程。
- 电路中含有受控电压源:列电路方程时,先用网孔电流将控制量表示出来,并暂时将受控电压源当做独立电压源,最后再将网孔电流表示的受控源电压移至方程的左边。
- 电路中含有电流源且无电阻直接与之并联,处理方法:①选取网孔电流时只让一个回路电流通过电流源,该网孔电流仅由电流源电流决定;②以电流源两端电压为变量,并且在每引入一个这样的变量时,增加一个网孔电流与电流源电流间的约束关系的方程。

**4) 节点电压法** 所谓的节点电压,是指两个节点电位之间的差值。引入节点电压法的目的和引入回路电流法的目的相同,都是为了简化分析和计算电路的步骤。

下面以图 1-13 所示电路为例,具体说明节点电压法的适用范围和解题步骤。

观察图 1-13 所示电路,该电路的结构特点是支路数多,回路数也多,但节点数较少。如果能像回路电流法省略掉 KCL 方程式那样,把 KVL 方程省略掉,只用 KCL 电流方程式进行解题,显然可大大减少该电路的方程式数目,从而达到简化分析步骤的目的。

寻求这种减少 KVL 方程数目的解题方法,要先从所有节点中找出其中的一个作为参考

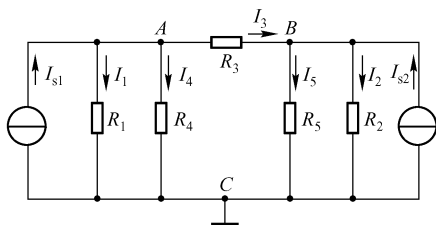


图 1-13 节点电压法电路举例

电位点，而任意一个节点上的电位都可以看做是该点与参考点之间的节点电压。

首先，选择  $C$  点作为电路参考点。由图 1-13 可知，恒流源  $I_{s1}$ 、电阻  $R_1$ 、电阻  $R_4$  的端电压等于  $A$  点电位  $V_A$ ；恒流源  $I_{s2}$ 、电阻  $R_2$  和电阻  $R_5$  的端电压等于  $B$  点电位  $V_B$ ；电压源  $U_{s3}$  与电阻  $R_3$  相串联的支路端电压则等于  $A$  点至  $B$  点的电位差  $V_A - V_B$ 。在图 1-13 中标示的各支路电流的参考方向下，根据欧姆定律可得：

$$I_1 = \frac{V_A}{R_1} \quad I_4 = \frac{V_A}{R_4} \quad I_2 = \frac{V_B}{R_2} \quad I_5 = \frac{V_B}{R_5} \quad I_3 = \frac{V_A - V_B}{R_3}$$

显然，只要求出各节点电位，由上述关系即可求出各支路电流。下面介绍如何求解节点电位。

假设电路中各节点电位已知，对电路中  $A$ 、 $B$  两个节点分别列写 KCL 方程式。

$$\text{对节点 } A: \frac{V_A}{R_1} + \frac{V_A}{R_4} + \frac{V_A - V_B}{R_3} = I_{s1}$$

$$\text{对节点 } B: \frac{V_B}{R_2} + \frac{V_B}{R_5} - \frac{V_A - V_B}{R_3} = I_{s2}$$

对两式进行整理可得：

$$\left( \frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_4} \right) V_A - \frac{1}{R_3} V_B = I_{s1} \quad (1)$$

$$\left( \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_5} \right) V_B - \frac{1}{R_3} V_A = I_{s2} \quad (2)$$

方程式①和②的左边是汇集到节点上的各未知支路电流，右边是已知电流。而左边第一项括号内各电导（电阻的倒数称为电导）之和称为自电导，自电导等于连接于本节点上所有支路的电导之和，恒为正值；左边第二项（或后几项）的电导为相邻节点与本节点之间公共支路上连接的电导，称为互电导，互电导总是取负值；各方程式右边则为汇集到本节点上的所有已知电流的代数和（仍然约定指向节点的电流取正，背离节点的电流取负）。由于这种解题形式是以节点电压为未知量，进而对电路进行分析计算的方法，因而称为节点电压法。

求得各节点电压后，还要根据待求量与节点电压之间的关系，最后得出各待求量。

节点电压法是以节点电位为未知量，进而对电路求解的分析方法，分析步骤如下。

- ① 选定参考节点。其余各节点与参考节点之间的电压就是待求的节点电压。
- ② 建立求解节点电压的 KCL 方程。一般可先算出各节点的自导、互导及汇集到本节点的已知电流代数和，然后直接代入节点电流方程。
- ③ 对方程式联立求解，得出各节点电压。
- ④ 选取各支路电流的参考方向，根据欧姆定律找出它们与各节点电压的关系，进而求解。

**5) 叠加定理** 叠加定理指出：在多个电源共同作用的线性电路中，任一支路的响应（即电流或电压），都可以看成是由每个激励（理想电压源或理想电流源）单独作用时在该支路中所产生的响应的代数和。

叠加定理体现了线性电路的基本特性——叠加性，是线性电路的一个重要定理。

应用叠加定理分析电路时，应注意以下几点。

- 叠加定理只适用于线性电路，对非线性电路不适用。在线性电路中，叠加定理也只能用来计算电流或电压，因为线性电路中的电压和电流响应，与电路激励之间的关系是一次函数的线性关系；功率与电路激励之间的关系是二次函数的非线性关系，因此不能用叠加定理进行分析和计算。
- 叠加时一般要注意使各电流、各电压分量的参考方向与原电流、电压的参考方向保持一致。若选取不一致，叠加时就要注意各电流、电压的正、负号；与原电流、电压的参考方向一致的电流、电压分量取正值，相反的取负值。
- 当某个独立源单独作用时，不作用的电压源应短路处理，不作用的电流源应开路处理。
- 叠加时，还要注意电路中所有电阻及受控源的连接方式都不能任意改动。

叠加定理不仅可以把一个复杂电路分解为多个简单电路，从而把复杂电路的分析计算变为简单电路的分析计算，它的重要性还在于：当线性电路中含有多种信号源激励时，它为研究响应与激励的关系提供了必要的理论根据和方法。线性电路的许多定理都可以从叠加定理导出。

**6) 戴维南定理** 在某些实际问题中，如果只求电路中某一支路的电流或电压，就不必列方程组求出所有支路的电流、电压，只需找出待求支路以外的二端网络的等值电路即可。对于无源二端网络来说，其等值电路是一个等值电阻。对于有源二端网络，其等值电路是什么？戴维南定理可以解决这个问题。

戴维南定理指出：任何一个含源线性二端网络，对外电路来说，都可以用一个电压源和电阻的串联组合来等效置换，此电压源的电压等于含源二端网络端口的开路电压，电阻等于该含源二端网络变为无源二端网络的入端电阻。上述电压源和电阻串联组成的电压源模型称为戴维南等效电路，该电阻称为戴维南等效电阻。

应用戴维南定理，对负载性质没有限定，关键是要要求出端口的开路电压及戴维南等效电阻。戴维南定理的求解步骤如下。

① 求开路电压：用电路的基本定律，或结合叠加原理。

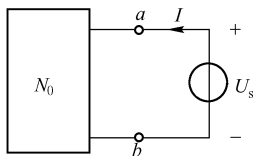
② 求戴维南等效电阻。

➤ 串并联法：令独立电源为零，根据网络结构，用串并联法求  $R_{eq}$ 。

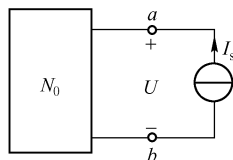
➤ 外加电源法：令网络中的独立电源为零，外加一电压源/电流源，用欧姆定律求  $R_{eq}$ 。

如图 1-14 (a) 所示，用外加电压源法，求得  $R_{eq} = \frac{U_s}{I}$ 。

如图 1-14 (b) 所示，用外加电流源法，求得  $R_{eq} = \frac{U}{I_s}$ 。



(a) 外加电压源电路



(b) 外加电流源电路

图 1-14 外加电源电路

应用戴维南定理要注意以下几个问题。

- 戴维南定理只适用于含源线性二端网络。因为戴维南定理是建立在叠加概念之上的，而叠加概念只能用于线性网络。
- 应用戴维南定理时，具有耦合的支路必须包含在网络 N 之内。
- 计算网络 N 的开路电压时，必须画出相应的电路，并标出开路电压的参考极性。
- 计算网络 N 的输出电阻时，也必须画出相应的电路。
- 在画戴维南等效电路时，等效电压源的极性应与开路电压相一致。
- 戴维南等效电路中等效的含义指的是，网络 N 用等效电路替代后，在连接端口 a、b 上，以及在 a、b 端口以外的电路中，电流、电压都没有改变。但在戴维南等效电路与被替代网络 N 中的内部情况，一般并不相同。



## 1.3 交流电路

交流电在人们的生产和生活中有着广泛的应用。在电网中由发电厂产生的电是交流电，输电线路上输送的也是交流电，各种交流电动机使用的仍然是交流电。交流电压、电流与直流电压、电流不同，它们的大小和方向随时间变化。常用的交流电是正弦交流电，即电压和电流的大小与方向按正弦规律变化，如图 1-15 所示为正弦交流电流。它的电压变换容易，输送和分配方便，供电性能好，效率高；交流电器结构简单、价格便宜、维修方便；从计算与分析的角度考虑，正弦周期函数是最简单的周期函数，测量与计算也比较容易，是分析一切非正弦周期函数的基础。

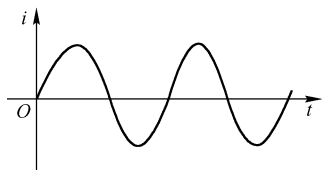


图 1-15 正弦交流电流

### 1. 正弦交流电的基本概念

**1) 正弦交流量的正方向** 正弦交流电路中的电压、电流及电动势，其大小和方向均随时间变化，数学表达式为：

$$\left. \begin{aligned} e &= E_m \sin(\omega t + \varphi_e) \\ v &= V_m \sin(\omega t + \varphi_v) \\ i &= I_m \sin(\omega t + \varphi_i) \end{aligned} \right\} \quad (1-16)$$

以  $v$  为例，其波形图如图 1-16 所示。在  $0 \sim t_1$  时间内若其实际正方向与参考正方向（箭头所标）相同，则在  $t_1 \sim t_2$  时间内，其实际正方向  $v'$  与参考正方向相反。因此，在分析交流电路时，不同瞬时交流量的比较是没有意义的。这也是其区别于直流电的基本特征。

**2) 正弦交流电的三要素** 式 (1-16) 是正弦交流量的瞬时值表达式，其中  $E_m$ 、 $V_m$ 、 $I_m$  称为正弦量的最大值或幅值； $\omega$  称为角频率； $\varphi_e$ 、 $\varphi_v$ 、 $\varphi_i$  称为初相位。如果已知幅值、角频率和初相位，则上述正弦

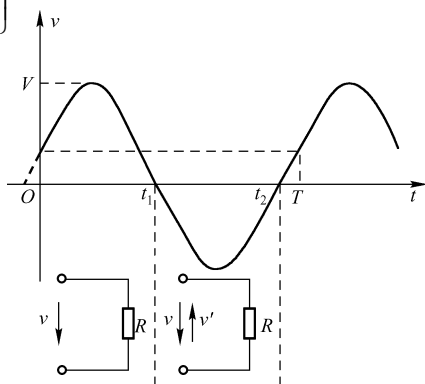


图 1-16 正弦量的波形与正方向

量就能唯一地确定, 所以称它们为正弦量的三要素。

(1) 最大值、瞬时值、有效值。最大值是反映正弦量变化幅度的, 又称为幅值或峰值, 规定用大写字母加下标  $m$  表示, 即  $E_m$ 、 $V_m$ 、 $I_m$ 。

瞬时值是正弦量任时刻的值, 规定用小写字母表示, 分别为  $e$ 、 $v$ 、 $i$ 。

平常所说的电压高低、电流大小及用电器上的标称电压或电流指的是有效值。有效值是由交流电在电路中做功的效果来定义的。叙述为: 交流电流  $i$  通过电阻  $R$  在一个周期  $T$  内产生的热量与直流电流  $I$  通过  $R$  在时间  $T$  内产生的热量相等时, 这个直流电流  $I$  的数值称为交流电流的有效值。数学表达为:

$$I^2 RT = \int_0^T i^2 R dt$$

则有效值表达式为:

$$I = \sqrt{\frac{1}{T} \int_0^T i^2 dt} \quad (1-17)$$

将式 (1-16) 的正弦量代入式 (1-17) 可得:

$$\left. \begin{aligned} I &= \frac{I_m}{\sqrt{2}} \\ V &= \frac{V_m}{\sqrt{2}} \\ E &= \frac{E_m}{\sqrt{2}} \end{aligned} \right\} \quad (1-18)$$

可见, 正弦交流量的最大值是其有效值的  $\sqrt{2}$  倍, 通常所说的交流电压 220V 是指有效值, 其最大值约为 311V。

(2) 周期、频率、角频率。反映交流电变化快慢的物理量是频率  $f$  (或周期  $T$ )。即交流电每秒变化的次数, 单位为赫兹 (Hz); 而周期为其交变一次所需的时间, 单位为秒 (s)。它们互为倒数的关系。目前世界各国电力系统的供电频率有 50Hz 和 60Hz 两种, 这种频率称为工业频率, 简称工频。不同技术领域中的频率要求是不一样的, 有的高达数千兆赫兹, 称为高频交流信号。

正弦交流量表达式中反映交流电变化快慢的特征量是角频率  $\omega$ , 一般正弦波形图中的横轴常用  $\omega t$  表示 (如图 1-17 所示), 可见  $\omega T = 2\pi$ , 则

$$\omega = \frac{2\pi}{T} = 2\pi f \quad (1-19)$$

角频率的单位是弧度/秒 (rad/s), 它的含义是正弦量每秒变化的弧度数, 或  $2\pi$  秒内交流量变化的周期数。它同样可以反映正弦量变化的快慢 (在交流发电机中,  $\omega$  又与发电机转动的角速度相联系)。

(3) 相位、初相位与相位差。任一瞬时的角度 ( $\omega t + \varphi$ ) 称为正弦量的相位角或相位, 它与交流量的瞬时值相联系。 $t=0$  时的相位角  $\varphi$  叫初相位角或初相位, 它是正弦量初始值大小的标志。

事实上, 初相位的大小与我们讨论它时所取的计时起点有关, 如果将图 1-17 中的计时

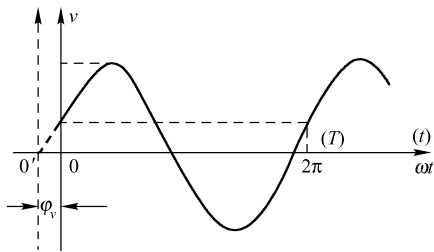


图 1-17 角频率与初相位的示意图



起点左移到图中虚线处, 则初相  $\varphi_v = 0$ 。当然, 初相位不同, 其起始值也不同。

在一个正弦电路中, 存在两个以上的正弦信号时, 一般不是同时达到最大值或零值的, 即它们之间存在着不同相位的问题。相位差是用来描述它们之间的先后关系的, 如

$$v = V_m \sin(\omega t + \varphi_v)$$

$$i = I_m \sin(\omega t + \varphi_i)$$

则它们的相位差为:

$$\varphi = (\omega t + \varphi_v) - (\omega t + \varphi_i) = \varphi_v - \varphi_i \quad (1-20)$$

可见, 同频正弦量的相位差也就是其初相位之差。

同频正弦量的相位差  $\varphi$  一般有以下三种情况。

①  $\varphi = \varphi_v - \varphi_i > 0$  (小于  $180^\circ$ ), 即  $\varphi_v > \varphi_i$ , 这种情况为先  $v$  后  $i$ , 称做  $v$  领先,  $i$  滞后, 如图 1-18 (a) 所示 (也可以哪个量先到最大值为参考来判别先后)。

②  $\varphi = \varphi_v - \varphi_i = 0$ , 即  $\varphi_v = \varphi_i$ , 称为同相位。同相位时两个正弦量同时增, 同时减, 同时到最大值, 同时过零, 如图 1-18 (b) 所示。

③  $\varphi = \varphi_v - \varphi_i = \pm \pi$ , 称为反相位, 如图 1-18 (c) 所示。

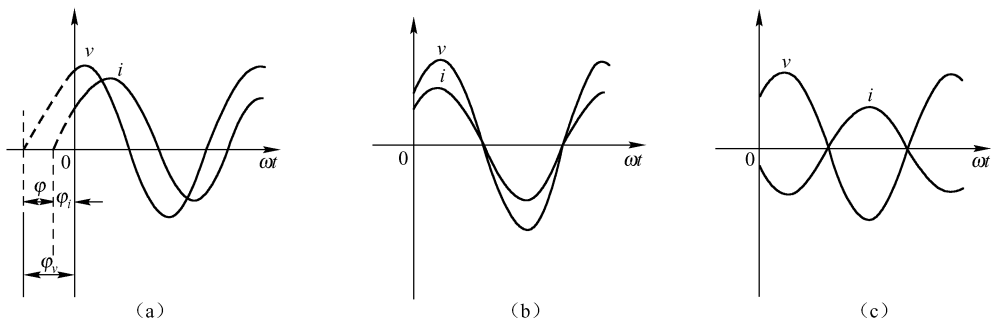


图 1-18 同频正弦量的相位差

需要说明的是, 虽然几个同频正弦量的相位都在随时间不停地变化, 但它们之间的相位差不变, 且与计时起点的选择无关。正是由于相位差的存在, 使得交流电路中出现许多新的物理现象; 同时也因相位差的存在, 使得交流电路问题的分析和计算要比直流电路复杂, 但内容更丰富。

## 2. 正弦交流电的向量表示法

用三角函数式或波形图来表达正弦量是最基本的表示方法, 但要用其进行电路分析与计算却比较烦琐。由于在正弦交流电路中一般使用的都是同频正弦量, 所以我们常用下面所述的向量图或向量表达式进行分析与计算, 这是电路理论中的基本表示法。

**1) 向量图** 向量图是能够确切表达正弦量三要素的简洁图示法。可以用复平面内长为幅值、以角速度  $\omega$  旋转的矢量来表示, 例如, 正弦电压  $v = V_m \sin(\omega t + \varphi_v)$  可为图 1-19 中左图的旋转矢量。此矢量大小为  $V_m$ , 以角速度  $\omega$  在复平面内旋转时, 任意时刻其矢端的纵坐标值与正弦波的瞬时值对应, 其与实轴的夹角即相位角为  $\omega t + \varphi_v$ , 为与空间矢量区别, 我们约定用大写字母上面加 “ $\cdot$ ” 表示, 如图中的  $\dot{V}_m$ 。

应用向量图分析正弦电压、电流问题时, 由于这些正弦量的频率相同 (即矢量的旋转

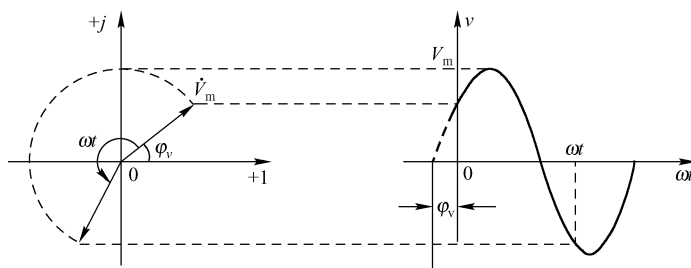


图 1-19 正弦量的向量图表示法

速度相同), 因而它们之间的相对位置在任何瞬间均不会改变。所以在分析时, 只需将它们当做不动量来处理即可, 这样不会影响分析的结果。此外, 工程计算中多用其有效值衡量大小, 故只需用有效值向量表示即可, 如

$$v_1 = V_{m1} \sin(\omega t + \varphi_1) \quad v_2 = V_{m2} \sin(\omega t + \varphi_2)$$

其向量图可简画为如图 1-20 (a) 所示, 其中  $V_1 = \frac{V_{m1}}{\sqrt{2}}$ 、 $V_2 = \frac{V_{m2}}{\sqrt{2}}$ 。若求电压  $v = v_1 + v_2$ ,

则由  $\dot{V}_1$ 、 $\dot{V}_2$  构成的平行四边形的对角线, 如图 1-20 (b) 所示。显而易见, 这样便可较方便地定出其和向量的有效值与初相位角, 且可表示为:

$$\dot{V} = \dot{V}_1 + \dot{V}_2$$

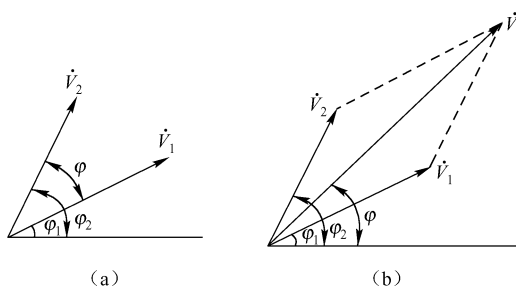


图 1-20 同频率正弦量的向量图与向量和

当然, 由向量图的计算结果变为正弦量, 只需将其值乘以  $\sqrt{2}$ , 加上旋转因子  $\omega t$  即可, 如

$$v = \sqrt{2} V \sin(\omega t + \varphi)$$

**2) 向量表达式 (复数符号法)** 用画向量图的方法可以清楚地表示所讨论的各正弦量间的相互关系, 也可通过作向量图求得所需结果, 但在实际使用时由于作图精度的限制, 特别是分析复杂电路时还是比较困难的。而向量的数学表达——复数符号法才是分析交流电路的一般方法。

若将图 1-19 中左图的向量用复数表示, 则为  $\dot{V}_m = |\dot{V}_m| e^{j(\omega t + \varphi_v)}$

根据欧拉公式:

$$V_m e^{j(\omega t + \varphi_v)} = V_m \cos(\omega t + \varphi_v) + j V_m \sin(\omega t + \varphi_v) \quad (1-21)$$

可见其虚部恰为我们所研究的正弦量, 即

$$v = V_m \sin(\omega t + \varphi_v) = I_m [V_m e^{j(\omega t + \varphi_v)}] \quad (1-22)$$

对于同频正弦量,  $\omega t$  可免写, 则其有效值向量可简记为:

$$\left. \begin{aligned} \dot{V} &= V e^{j\varphi_v} = V \angle \varphi_v \\ \dot{I} &= I e^{j\varphi_i} = I \angle \varphi_i \\ \dot{E} &= E e^{j\varphi_e} = E \angle \varphi_e \end{aligned} \right\} \quad (1-23)$$

这种表示叫向量的极坐标表示法。需要说明的是, 只有在电路与电工类书籍中才这样表达; 并且, 只有用复数表示的正弦量才叫向量, 用复数表示的其他量不能叫向量。

借助于向量的复数表示, 结合向量图, 同频正弦量的分析与计算可以一步求得其大小(幅值)与初相位(辐角), 方便多了。

当然, 求得其大小与相位角后, 还需将其再写为正弦形式, 即取虚部、乘以  $\sqrt{2}$ 、加上旋转因子  $\omega t$ 。



## 1.4 思考与练习

1. 什么是理想电路元件? 如何理解“理想”二字在实际电路中的含义? 什么是电路模型?
2. 一只“ $100\Omega$ 、 $100W$ ”的电阻与  $120V$  电源相串联, 至少要串入多大的电阻  $R$  才能使该电阻正常工作? 电阻  $R$  上消耗的功率又为多少?
3.  $I_A$ 、 $I_B$ 、 $I_C$  的参考方向如图 1-21 中所示, 这三个电流有无可能都是正值?
4. 在应用 KCL 定律解题时, 为什么要先约定流入、流出节点电流的正、负? 计算结果电流为负值说明了什么问题?
5. 求图 1-22 中电流  $I_5$  的数值, 已知  $I_1 = 4A$ ,  $I_2 = -2A$ ,  $I_3 = 1A$ ,  $I_4 = -3A$ 。
6. 在图 1-23 所示电路中, 已知  $I_1 = -2A$ ,  $I_2 = 6A$ ,  $I_3 = 3A$ ,  $I_5 = -3A$ , 参考方向如图所示。求元件 4 和元件 6 中的电流。

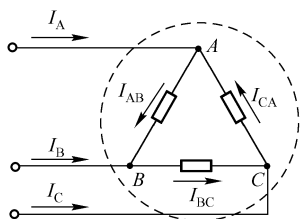


图 1-21 习题 1-3 图

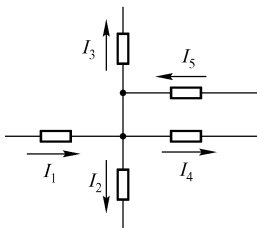


图 1-22 习题 1-5 图

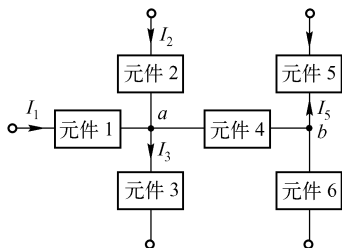


图 1-23 习题 1-6 图

7. 在图 1-24 所示电路中, 利用 KVL 求解图示电路中的电压  $U$ 。
8. 如图 1-25 所示, 列写支路电流方程 (电路中含有理想电流源)。
9. 用节点电压法求解图 1-26 所示电路中  $50k\Omega$  电阻中的电流  $I$ 。
10. 某浮充供电电路如图 1-27 所示, 整流器直流输出电压  $U_{s1} = 250V$ , 等效内阻  $R_{s1} = 1\Omega$ , 浮充蓄电池组的电压值  $U_{s2} = 239V$ , 内阻  $R_{s2} = 0.5\Omega$ , 负载电阻  $R_L = 30\Omega$ , 分别用支路电流法和回路电流法求解各支路电流、负载端电压及负载上获得的功率。

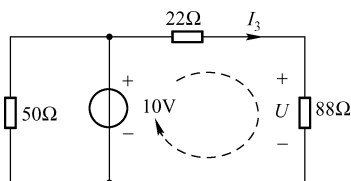


图 1-24 习题 1-7 图

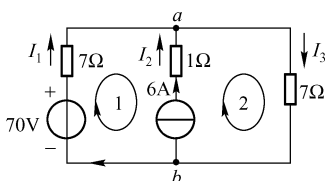


图 1-25 习题 1-8 图

11. 求图 1-28 所示电路中的电流  $I_2$ 。

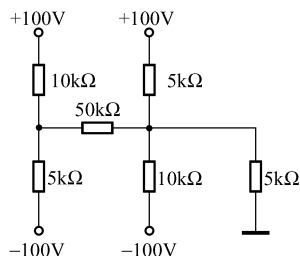


图 1-26 习题 1-9 图

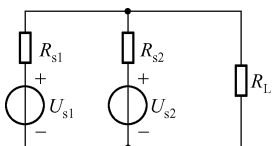


图 1-27 习题 1-10 图

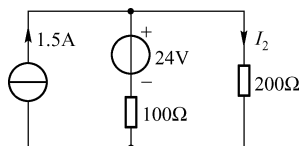


图 1-28 习题 1-11 图

12. 叠加定理的适用范围是什么？它是否仅适用于直流电路而不适用于交流电路的分析和计算？

13. 应用戴维南定理求解图 1-29 中  $5\Omega$  电阻上的电压  $U$ 。

14. 应用戴维南定理求解图 1-30 所示电路中的电流  $I$ ，再用叠加定理进行校验。

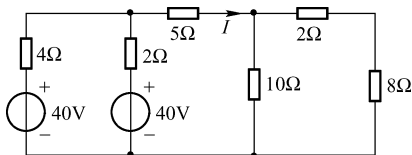


图 1-29 习题 1-13 图

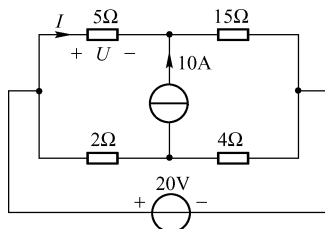


图 1-30 习题 1-14 图

15. 电流和电压可以应用叠加定理进行分析和计算，功率为什么不行？

16. 如何求解戴维南等效电路的电压源  $U_s$  及内阻  $R_0$ ？该定理的物理实质是什么？

17. 一个正弦电流的初相位  $\varphi = 15^\circ$ ， $t = \frac{T}{4}$ ， $i(t) = 0.5\text{ A}$ ，试求该电流的有效值  $I$ 。

18. 已知  $i_1 = 5\sqrt{2} \sin(\omega t + 30^\circ)\text{ A}$ ， $i_2 = 10\sqrt{2} \sin(\omega t + 60^\circ)\text{ A}$ ，完成：（1）求  $\dot{I}_1$ 、 $\dot{I}_2$ ；（2） $\dot{I}_1 + \dot{I}_2$ ；（3）求  $i_1 + i_2$ ；（4）作向量图。

## 第2章 电子电路设计基础

电子系统是指电子元件和电子单元电路相互连接、相互作用而形成的能按特定的控制信号去执行所设想的功能的电路整体。按处理信号的不同,电子系统一般可分为模拟电子系统、数字电子系统和数字模拟混合系统。本章将介绍一些常用的模拟电路和数字电路,主要包括信号放大、信号滤波、组合逻辑电路和时序逻辑电路等。



### 2.1 信号放大

工业生产和日常生活中需要将微弱变化的电信号放大几百倍、几千倍甚至几十万倍之后去带动执行机构,对生产设备进行测量、控制或调节,完成这一任务的电路称为放大电路,简称放大器。放大器是任何一台现代测量仪器不可缺少的基本电路。越灵敏的仪器,越需要高增益、高性能的放大器。根据实际仪器功能和要求的不同,对放大器也有这样或那样的性能要求,如增益的高低、频带的宽窄、输入阻抗的高低,等等。实际上,放大器的参数远不止这些,还有许许多多的参数来表征放大器,如非线性放大器、程控放大器、差动放大器、微功耗放大器……所以,放大器的种类举不胜举。往常,通用运算放大器是设计工程师们的“万金油”,不管什么样的放大器都用通用运算放大器来设计。

#### 1. 运算放大器的电路模型

运算放大器是一种多端电子器件,最早开始应用于1940年,首先应用于模拟计算机上,作为基本运算单元,可以完成加减、积分和微分、乘除等数学运算。自1960年后,随着半导体集成工艺的发展,运算放大器逐步集成化,大大降低了成本,其应用远远超出了模拟计算机的界限,在信号运算、信号处理、信号测量及波形产生等方面获得广泛应用。

- 信号的运算电路:比例、加、减、对数、指数、积分、微分等运算。
- 信号的处理电路:有源滤波器、精密整流电路、电压比较器、采样-保持电路。
- 信号的发生电路:产生方波、锯齿波等波形。

**1) 运算放大器的符号** 运算放大器是由具有高放大倍数的直接耦合放大电路组成的半导体多端实际元件,简称运放。在本章中所讲到的“运算放大器”,是指实际运算放大器的电路模型——一种四端元件,其符号如图2-1所示。

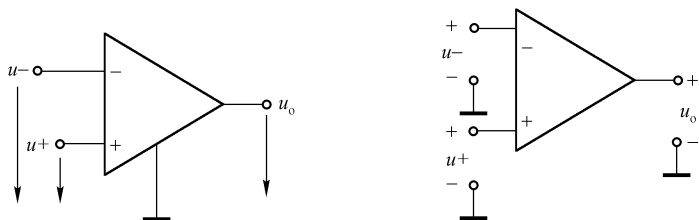


图2-1 运放的符号

在新国标中, 运算放大器及理想运算放大器的符号如图 2-2 所示。



图 2-2 运放的新国标符号

- 同相与反相输入端: 运算放大器符号中的“+”、“-”表示运算放大器的同相输入端和反相输入端, 即当输入电压加在同相输入端和公共端之间时, 输出电压和输入电压两者的实际方向相对于公共端来说相同; 反之, 当输入电压加在反相输入端和公共端之间时, 输出电压和输入电压两者的实际方向相对于公共端来说相反。其意义并不是电压的参考方向。
- 公共端: 在运算放大器中, 公共端往往取定为接地端——电位为零。实际中, 电子线路中的接地端常常取多条支路的汇合点、仪器的底座或机壳等, 输入电压、输出电压都以之为参考点。有时, 电路中并不画出该接地端, 但计算时要注意它始终存在。

**2) 运算放大器的模型** 由图 2-3 所示运算放大器的这一模型, 可以通过将运放等效为一个含有受控源的电路, 从而进行分析计算。

## 2. 集成运放主要技术指标及理想集成运放特性

**1) 集成运放主要技术指标** 为了描述集成运放的性能, 方便正确地选用集成运放, 必须明确它的参数意义。下面介绍几个主要参数。

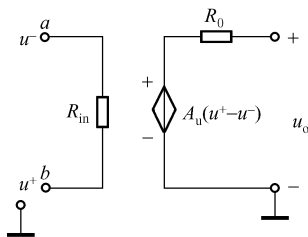


图 2-3 运放的电路模型

(1) 开环差模电压放大倍数  $A_{od}$ 。它是指集成运放无反馈电路情况下的差模电压放大倍数, 一般希望这个值越大越好, 理想集成运放的开环差模电压放大倍数为无穷大。

(2) 差模输入电阻  $R_{id}$ 。差模输入电阻  $R_{id}$  是指运放两个输入端在输入差模信号时所呈现的阻抗, 其值越大, 对上一级放大器或信号源的影响就越小。

(3) 输入失调电压  $U_{io}$  及输入失调电压温漂  $\frac{dU_{io}}{dT}$ 。由于实际集成运放的内部电路不可能做到完全对称, 所以, 即使输入电压为零, 输出电压也有一定的电压, 为了使输出电压为零, 在输入端加上一个补偿电压, 把这个补偿电压称为输入失调电压  $U_{io}$ 。其值越小, 表示运放电路的对称性越好。输入失调电压随温度变化而变化, 通常将输入失调电压对温度的平均变化率称为输入失调电压温漂, 用  $\frac{dU_{io}}{dT}$  表示。这个值越小, 说明运放的性能越好。

(4) 输入失调电流  $I_{io}$  及输入失调电流温漂  $\frac{dI_{io}}{dT}$ 。由于运放内部电路的不对称性, 两个输入端的基极静态电流也就不相等。这两个基极静态电流之差称为输入失调电流  $I_{io}$ , 其值越小越好。输入失调电流同样受温度影响, 把输入失调电流对温度的平均变化率称为输入失调电流温漂, 用  $\frac{dI_{io}}{dT}$  表示。

(5) 转换速率  $S_R$ 。转换速率  $S_R$  是指集成运放输入幅度较大的阶跃信号时, 输出电压随时间的最大变化速率。定义为:

$$S_R = \left| \frac{du_o(t)}{dt} \right|_{\max} \quad (2-1)$$

$S_R$  反映了运放输出电压对高速变化的输入信号的响应能力, 只有当输入信号的变化率小于运放的  $S_R$  时, 输出电压才会随输入电压线性变化。 $S_R$  越大, 运放的高频特性越好。一般运放的  $S_R$  为  $(0.5 \sim 100) \text{ V/s}$  数量级, 高速运放甚至可达  $1000 \text{ V/s}$  以上。

(6)  $-3\text{dB}$  带宽  $f_H$ 。把电压放大倍数下降到中频时的  $0.707$  倍 (用增益表示恰好是下降  $3\text{dB}$ ) 所对应的频率  $f_H$  称为截止频率, 也叫  $-3\text{dB}$  带宽。

除了上述几个参数以外, 还有共模抑制比、输出电阻、最大输出电压等大家熟悉的参数, 在这里就不重复介绍了。

在分析集成运放时, 常常将它看成理想运算放大器, 可以简化分析过程, 并且计算产生的误差也很小。在没有特别说明的情况下, 都把它看成理想运算放大器。通常把理想运算放大器的各项技术参数都理想化, 其主要参数如下:

- 开环差模电压放大倍数  $A_{od} = \infty$ ;
- 差模输入电阻  $r_{id} = \infty$ ;
- 输出电阻  $R_{od} = 0$ ;
- 共模抑制比  $K_{CMR} = \infty$ ;
- 频带宽度和转换速率无穷大。

**2) 集成运放的传输特性** 集成放运的传输特性是指集成运放的输出与输入关系的特性。它可分为线性区和非线性区, 如图 2-4 所示为集成运放的传输特性。实际上, 运算放大器是一种单向器件, 即输出电压受输入电压的控制, 而输入电压并不受输出电压的控制。由其输入/输出关系可以看出, 运算放大器的线性放大部分很窄, 当输入电压很小时, 运算放大器的工作状态就已经进入饱和区, 输出值开始保持不变。下面分别分析理想运算放大器工作在线性区和非线性区的特点。

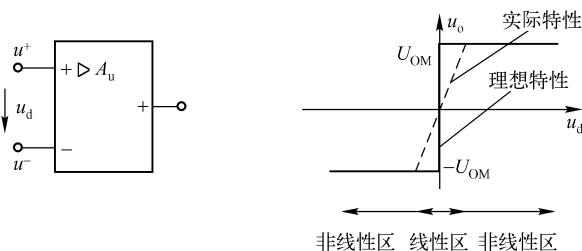


图 2-4 集成运放的传输特性

**注意:** 运算放大器可以工作在线性区, 也可以工作在饱和区, 但分析的方法不一样。当运算放大器工作在线性区时是一个线性放大元件, 本章把运算放大器的工作范围限制在线性区。

理想运放工作在线性区时, 其输出电压与输入电压成线性放大关系, 即

$$u_o = A_{od}(u^+ - u^-) = A_{od}u_d \quad (2-2)$$

式中,  $A_{od}$  为运算放大器的放大倍数。

为了使运放工作在线性区, 一般通过引入负反馈的方法来实现。在线性工作区, 理想运放有两个重要特点。

(1) 两输入端电位相等。由于理想运放的开环放大倍数  $A_{od} = \infty$ ，而  $u_o$  是一个有限值，所以：

$$u_+ = u_-$$

两输入端就像“短路”一样，但它们又不是真正短路，所以称为“虚短”。

(2) 两输入电流为零。由于理想运放的输入电阻  $r_{id} = \infty$ ，因此，两输入端均没有电流输入，即

$$i_+ = i_- = 0$$

两输入端就像“断路”一样，但它们又不是真正断路，所以称为“虚断”。

利用“虚短”和“虚断”这两个重要特点，在分析和处理运放电路时，使过程变得更加简单。

**3) 非线性区** 当输入信号过大时，运放就会工作在非线性区，如图 2-4 所示，这时输出电压不再随输入电压线性增长，而以固定的电压 ( $U_{OM}$  或  $-U_{OM}$ ) 输出。

理想运放工作在非线性区时也有两个重要特点。

(1) 理想运放的输出电压  $u_o$ ，以两种可能的定值输出。

当输入电压  $u_+ > u_-$  时， $u_o = U_{OM}$ ；

当输入电压  $u_+ < u_-$  时， $u_o = -U_{OM}$ 。

在非线性区内，运放的差模输入电压  $u_+ - u_-$  可能较大，即  $u_+ \neq u_-$ 。此时，“虚短”现象不复存在， $u_+ = u_-$  只是作为一种暂态。

(2) 理想运放的输入电流等于零。在非线性区，虽然运放两个输入端的电位不等，但因为理想运放的  $V_{id} = \infty$ ，故仍可认为理想运放的输入电流等于零。即  $i_+ = i_- = 0$ 。此时，“虚断”仍然成立。

实际运放的传输特性如图 2-4 中虚线所示，因为集成运放的  $A_{od}$  值通常很大，所以线性放大范围很小。

由此可见，理想集成运放工作在线性区和非线性区的特点各不相同。因此，在分析具体的集成运放应用电路时，要先判断它工作在哪个区。

### 3. 集成运算放大器的分类及应用

**1) 集成运算放大器的分类** 按照集成运算放大器的参数来分，集成运算放大器可分为如下几类。

(1) 通用型运算放大器。通用型运算放大器是以通用为目的而设计的。这类器件的主要特点是价格低廉，产品量大、面广，其性能指标适合于一般性使用。例如， $\mu A741$ （单运放）、LM358（双运放）、LM324（四运放）及以场效应管为输入级的 LF356 都属于此种类型。它们是目前应用最为广泛的集成运算放大器。

(2) 高阻型运算放大器。这类集成运算放大器的特点是差模输入阻抗非常高，输入偏置电流非常小，一般  $R_{id} > (10^9 \sim 10^{12}) \Omega$ ， $I_{ib}$  为几皮安到几十皮安。实现这些指标的主要措施是利用场效应管高输入阻抗的特点，用场效应管组成运算放大器的差分输入级。用 FET 作输入级，不仅输入阻抗高，输入偏置电流小，而且具有高速、宽带和低噪声等优点，但输入失调电压较大。常见的集成器件有 LF355、LF347（四运放）及更高输入阻抗的 CA3130、CA3140 等。

(3) 低温漂型运算放大器。在精密仪器、弱信号检测等自动控制仪表中，总是希望运



算放大器的失调电压要小且不随温度的变化而变化。低温漂型运算放大器就是为此而设计的。目前常用的高精度、低温漂运算放大器有 OP07、OP27、AD508 及由 MOSFET 组成的斩波稳零型低漂移器件 ICL7650 等。

(4) 高速型运算放大器。在快速 A/D 和 D/A 转换器、视频放大器中, 要求集成运算放大器的转换速率  $S_R$  一定要高, 单位增益带宽 BWG 一定要足够大, 像通用型集成运放是不能适合于高速应用的场合的。高速型运算放大器主要特点是具有高的转换速率和宽的频率响应。常见的运放有 LM318、 $\mu A715$  等, 其  $S_R = 50 \sim 70 \text{ V}/\mu\text{s}$ ,  $\text{BWG} > 20 \text{ MHz}$ 。

(5) 低功耗型运算放大器。由于电子电路集成化的最大优点是能使复杂电路小型轻便, 所以随着便携式仪器应用范围的扩大, 必须使用低电源电压供电、低功率消耗的运算放大器与之相适用。常用的运算放大器有 TL-022C、TL-060C 等, 其工作电压为  $\pm 2 \sim \pm 18 \text{ V}$ , 消耗电流为  $50 \sim 250 \mu\text{A}$ 。目前有的产品功耗已达  $\mu\text{W}$  级, 例如, ICL7600 的供电电源为  $1.5 \text{ V}$ , 功耗为  $10 \text{ mW}$ , 可采用单节电池供电。

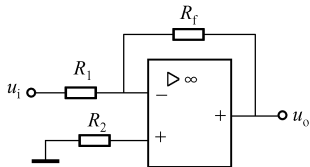
(6) 高压大功率型运算放大器。运算放大器的输出电压主要受供电电源的限制。在普通的运算放大器中, 输出电压的最大值一般仅几十伏, 输出电流仅几十毫安。若要提高输出电压或增大输出电流, 集成运放外部必须加辅助电路。高压大电流集成运算放大器外部无须附加任何电路, 即可输出高电压和大电流。例如, D41 集成运放的电源电压可达  $\pm 150 \text{ V}$ ,  $\mu A791$  集成运放的输出电流可达  $1 \text{ A}$ 。

(7) 可编程控制运算放大器。在仪器仪表的使用过程中都会涉及量程的问题。为了得到固定电压的输出, 必须改变运算放大器的放大倍数。例如, 有一运算放大器的放大倍数为 10 倍, 输入信号为  $1 \text{ mV}$  时, 输出电压为  $10 \text{ mV}$ , 当输入电压为  $0.1 \text{ mV}$  时, 输出就只有  $1 \text{ mV}$ , 为了得到  $10 \text{ mV}$  就必须改变放大倍数为 100。程控运放就是为了解决这一问题而产生的。如 PGA103A, 通过控制 1、2 脚的电平来改变放大的倍数。

**2) 集成运算放大器的应用** 集成运放最早是用于信号的运算, 它可以对信号进行加、减、乘、除、积分、微分、对数和反对数等运算, 所以称为运算放大器。但是随着集成运放的不断发展, 技术日益完善, 成本不断降低, 目前已经造出很多特殊功能的集成运放电路。如今, 集成运放已经应用于电子技术的各个领域, 不仅对信号进行运算, 而且可以对信号进行比较处理, 可以产生正弦波和非正弦波信号, 成为电子系统里的一个基本功能单元。

(1) 集成运算放大器的线性应用。线性应用是指运算放大器工作在线性状态, 即输出电压与输入电压呈线性关系, 主要用于实现对各种模拟信号进行比例、求和、积分、微分等数学运算, 以及有源滤波、采样保持等信号处理工作, 分析方法是应用“虚断”和“虚短”这两条分析依据。线性应用的条件是必须引入深度负反馈。集成运算放大器线性应用的基本电路及输出电压与输入电压的关系 (电压传输关系) 如表 2-1 所示。

表 2-1 集成运算放大器线性应用的基本电路及输出电压与输入电压的关系

名 称	电 路	电压传输关系	说 明
反相比例运算		$u_o = -\frac{R_f}{R_1} u_i$ $R_2 = R_1 // R_f$	电压并联负反馈 $u_- = u_+ = 0$ $R_2$ 为平衡电阻

续表

名 称	电 路	电压传输关系	说 明
同相比例运算		$u_o = \left(1 + \frac{R_f}{R_1}\right) u_i$ $R_2 = R_1 // R_f$	电压串联负反馈 $u_- = u_+ = u_i$ $R_2$ 为平衡电阻
电压跟随器		$u_o = u_i$	电压串联负反馈 $u_- = u_+ = u_i$
反相加法运算		$u_o = -\left(\frac{R_f}{R_1} u_{i1} + \frac{R_f}{R_2} u_{i2}\right)$ $R_3 = R_1 // R_2 // R_f$	电压并联负反馈 $u_- = u_+ = 0$ $R_2$ 为平衡电阻
减法运算		$u_o = -\frac{R_f}{R_1} u_{i1} + \left(1 + \frac{R_f}{R_1}\right) \frac{R_3}{R_2 + R_3} u_{i2}$ <p>当 <math>R_f = R_1</math>, <math>R_3 = R_2</math> 时</p> $u_o = \frac{R_f}{R_1} (u_{i2} - u_{i1})$ $R_2 // R_3 = R_1 // R_f$	$R_f$ 对 $u_{i1}$ 电压并联负反馈, 对 $u_{i2}$ 电压串联负反馈 $u_- = u_+ = 0$ 运用叠加定理分析
积分运算		$u_o = -\frac{1}{RC} \int u_i dt$ $R_1 = R$	电压并联负反馈 $u_- = u_+ = 0$ $R_1$ 为平衡电阻
微分运算		$u_o = -RC \frac{du_i}{dt}$ $R_1 = R$	电压并联负反馈 $u_- = u_+ = 0$ $R_1$ 为平衡电阻
有源低通滤波器		$\frac{U_o}{U_i} = \frac{1 + \frac{R_f}{R_1}}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^2}}$	电压并联负反馈 $u_- = u_+ = 0$ $\omega_c = \frac{1}{RC}$
有源高通滤波器		$\frac{U_o}{U_i} = \frac{1 + \frac{R_f}{R_1}}{\sqrt{1 + \left(\frac{\omega_c}{\omega}\right)^2}}$	电压并联负反馈 $u_- = u_+ = 0$ $\omega_c = \frac{1}{RC}$

(2) 集成运算放大器的非线性应用。非线性应用是指运算放大器工作在饱和（非线性）状态，输出为正的饱和电压或负的饱和电压，即输出电压与输入电压是非线性关系，主要用于实现电压比较、非正弦波发生等，分析依据是  $i_+ = i_- = 0$ ， $u_+ > u_-$  时  $u_o = U_{OM}$ ， $u_+ < u_-$  时  $u_o = -U_{OM}$ ，其中  $u_+ = u_-$  为转折点。线性应用的条件是工作在开环状态或引入正反馈。集成运算放大器非线性应用的基本电路及输出电压与输入电压的关系如表 2-2 所示。

表 2-2 集成运算放大器非线性应用的基本电路及输出电压与输入电压的关系

名 称	电 路	电压传输关系	说 明
任意电压比较器	反相输入 		$u_- = u_i, u_+ = U_R$ $u_i < U_R$ 时, $u_o = U_{OM}$ $u_i > U_R$ 时, $u_o = -U_{OM}$ $U_R = 0$ 时为过零比较器
	同相输入 		$u_+ = u_i, u_- = U_R$ $u_i > U_R$ 时, $u_o = U_{OM}$ $u_i < U_R$ 时, $u_o = -U_{OM}$ $U_R = 0$ 时为过零比较器
限幅电压比较器	反相输入 		$u_- = u_i, u_+ = U_R$ $u_i < U_R$ 时, $u_o = U_Z$ $u_i > U_R$ 时, $u_o = -U_Z$ $U_R = 0$ 时为过零比较器
	同相输入 		$u_+ = u_i, u_- = U_R$ $u_i > U_R$ 时, $u_o = U_Z$ $u_i < U_R$ 时, $u_o = -U_Z$ $U_R = 0$ 时为过零比较器
滞回比较器			上门限电压 $u_{H1} = \frac{R_2}{R_2 + R_f} U_{OM}$ 下门限电压 $u_{H2} = -\frac{R_2}{R_2 + R_f} U_{OM}$ 回差电压 $u_H = u_{H1} - u_{H2} = \frac{2R_2}{R_2 + R_f} U_{OM}$



## 2.2 信号滤波

在对拾取的微小信号放大的过程中，不可避免会有干扰和噪声。滤波是抑制或消除噪声的主要手段之一，而滤波器实现了对信号的滤波功能。

滤波器的功能就是允许某一部分频率的信号顺利地通过，而另外一部分频率的信号则受到较大的抑制，它实质上是一个选频电路。滤波器中，把信号能够通过的频率范围称为通频带或通带；反之，信号受到很大衰减或完全被抑制的频率范围称为阻带；通带和阻带之间的分界频率称为截止频率。理想滤波器在通带内的电压增益为常数，在阻带内的电压增益为

零；实际滤波器的通带和阻带之间存在一定频率范围的过渡带。

滤波器的分类如下。

(1) 按所处理的信号分为模拟滤波器和数字滤波器两种。

(2) 按所通过信号的频段分为低通滤波器、高通滤波器、带通滤波器和带阻滤波器 4 种，如图 2-5 所示。

- 低通滤波器：它允许信号中的低频或直流分量通过，抑制高频分量或干扰和噪声。
- 高通滤波器：它允许信号中的高频分量通过，抑制低频或直流分量。
- 带通滤波器：它允许一定频段的信号通过，抑制低于或高于该频段的信号、干扰和噪声。
- 带阻滤波器：它抑制一定频段内的信号，允许该频段以外的信号通过。

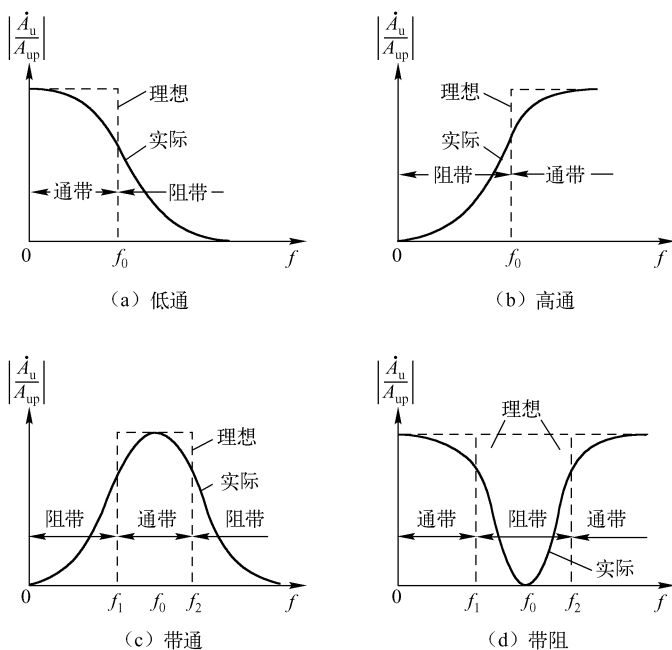


图 2-5 4 种滤波电路的幅频特性示意图

(3) 按所采用的元器件分为无源滤波器和有源滤波器两种。

- 无源滤波器：仅由无源元件 ( $R$ 、 $L$  和  $C$ ) 组成的滤波器，它是利用电容和电感元件的电抗随频率的变化而变化的原理构成的。这类滤波器的优点是：电路比较简单，不需要直流电源供电，可靠性高。缺点是：电路增益小，最大为 1；通带内的信号有能量损耗，负载效应比较明显，使用电感元件时容易引起电磁感应，当电感  $L$  较大时滤波器的体积和重量都比较大，在低频域不适用。
- 有源滤波器：由无源元件（一般用  $R$  和  $C$ ）和有源器件（如集成运算放大器）组成。这类滤波器的优点是：通带内的信号不仅没有能量损耗，而且还可以放大；带负载能力强，多级相联时相互影响很小，利用级联的简单方法很容易构成高阶滤波器，并且滤波器的体积小、重量轻，不需要磁屏蔽（由于不使用电感元件）。缺点是：通带范围受有源器件（如集成运算放大器）的带宽限制，需要直流电源供电，可靠性不如无源滤波器高，在高压、高频、大功率的场合不适用。

## 1. 滤波器的主要特性指标

1) 增益与衰耗 滤波器在通带内的增益为  $K_p$ ，并非常数。

(1) 对低通滤波器，通带增益  $K_p$  一般指  $f=0$  时的增益；高通指  $f \rightarrow \infty$  时的增益；带通

则指中心频率  $f_0$  处的增益。

(2) 对带阻滤波器, 应给出阻带衰耗, 衰耗定义为增益的倒数, 即衰耗  $= 1/K_p$ 。

(3) 通带增益变化量  $\Delta K_p$  (常称为通带波纹) 指通带内各点增益的最大变化量, 若  $\Delta K_p$  以 dB 为单位, 则指增益 dB 值的变化量。

2) 特征角频率  $\omega_n$  和特征频率  $f_n$  它只与滤波用的电阻和电容元件的参数有关, 通常:  $\omega_n = 1/RC$ ,  $f_n = \frac{1}{2\pi RC}$ 。

对于带通 (带阻) 滤波器, 称为带通 (带阻) 滤波器中心角频率的  $\omega_0$  或中心频率  $f_0$  是通带 (阻带) 内电压增益最大 (最小) 点的频率。

3) 截止角频率  $\omega_c$  和截止频率  $f_0$  它是电压增益下降到  $|A_0|/\sqrt{2}$  时所对应的角频率。必需要注意,  $\omega_c$  不一定等于  $\omega_n$ 。带通和带阻滤波器有两个  $\omega_c$ , 即  $\omega_L$  和  $\omega_H$ 。

4) 通带 (阻带) 宽度 BW 指带通 (带阻) 滤波器两个  $\omega_c$  的差值, 即  $BW = \omega_H - \omega_L$ 。

5) 阻尼系数  $\alpha$  和等效品质因数  $Q$  阻尼系数  $\alpha$  表征了滤波器对角频率为  $\omega_0$  的信号的阻尼作用, 是滤波器中表示能量衰减的一项指标。阻尼系数  $\alpha$  的倒数称为品质因素  $Q$ , 是评价带通和带阻滤波器频率选择性的一个重要指标。

对低通和高通滤波器而言,  $Q$  值等于  $\omega = \omega_n$  时滤波器电路电压增益的模与通带增益之比, 即  $Q = |A(j\omega_n)/A_0|$ ; 对带通 (带阻) 滤波器而言,  $Q$  值等于中心角频率与通带 (阻带) 宽度 BW 之比, 即  $Q = \omega_0/BW$ 。

6) 灵敏度 滤波电路由许多元件构成, 每个元件参数值的变化都会影响滤波器的性能。滤波器某一性能指标  $y$  对某一元件参数  $x$  变化的灵敏度记做  $S_x^y$ , 定义为:  $S_x^y = \frac{dy/y}{dx/x}$ 。该灵敏度与测量仪器或电路系统灵敏度不是一个概念, 该灵敏度越小, 标志着电路容错能力越强, 稳定性也越高。

7) 群时延函数 当滤波器的幅频特性满足设计要求时, 为保证输出信号失真度不超过允许范围, 对其相频特性  $\varphi(\omega)$  也应提出一定的要求。在滤波器设计中, 常用群时延函数  $\tau(\omega) = \frac{d(\varphi\omega)}{d\omega}$  评价信号经滤波后的相位失真程度。群时延函数  $\tau(\omega)$  越接近常数, 信号相位失真越小。

## 2. 滤波器的传递函数与频率特性

1) 模拟滤波器的传递函数 模拟滤波器的特性可由传递函数来描述。传递函数是输出与输入信号的拉氏变换之比。

$$H(s) = \frac{U_o(s)}{U_i(s)} = \frac{b_ms^m + b_{m-1}s^{m-1} + \cdots + b_1s + b_0}{a_ns^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} \quad (2-3)$$

任意个互相隔离的线性网络级联后, 总传递函数等于各网络传递函数的乘积。即任何复杂的滤波网络, 均可由若干个简单的一阶与二阶滤波电路级联构成。二阶滤波器是基本的滤波器。

二阶滤波器传递函数的一般形式为:

$$H(s) = \frac{b_0S^2 + b_1S + b_2}{S^2 + a_1S + a_2} \quad (2-4)$$

**2) 频率特性** 模拟滤波器的传递函数  $H(s)$  表达了滤波器的输入与输出之间的传递关系。若将传递函数中的  $s \rightarrow j\omega$ , 则  $H(j\omega)$  表示滤波器对正弦信号输入时的稳态响应。在正弦信号输入下的输出信号随频率变化的关系, 称为滤波器的频率特性函数, 简称频率特性。

频率特性  $H(j\omega)$  是一个复函数, 其幅值  $A(\omega)$  表示输出信号与输入信号的幅值比, 称为幅频特性; 其幅角  $\varphi(\omega)$  表示输出信号相对于输入信号的相位差, 称为相频特性。

理想滤波器:  $A(\omega) = \text{常数}$ ,  $\varphi(\omega) = \omega t_0 \propto \omega$ 。

### 3. 有源滤波器的设计

由 RC 元件与运算放大器组成的滤波器称为 RC 有源滤波器, 其功能是让一定频率范围内的信号通过, 抑制或急剧衰减此频率范围以外的信号。RC 有源滤波器可用在信息处理、数据传输、抑制干扰等方面, 但因受运算放大器频带的限制, 这类滤波器主要用于低频范围。根据对频率范围的选择不同, 其可分为低通 (LPF)、高通 (HPF)、带通 (BPF) 和带阻 (BEF) 4 种滤波器。

具有理想幅频特性的滤波器是很难实现的, 只能用实际的幅频特性去逼近理想的。一般来说, 滤波器的幅频特性越好, 其相频特性越差, 反之亦然。滤波器的阶数越高, 幅频特性衰减的速率越快, 但 RC 网络的节数越多, 元件参数计算越烦琐, 电路调试越困难。任何高阶滤波器均可用较低的二阶 RC 有源滤波器通过级联实现。

**1) 低通滤波器 (LPF)** 低通滤波器是用来通过低频信号、衰减或抑制高频信号的。如图 2-6 (a) 所示, 为典型的二阶有源低通滤波器。它由两级 RC 滤波环节与同相比例运算电路组成, 其中第一级电容  $C$  接至输出端, 引入适量的正反馈, 以改善幅频特性。图 2-6 (b) 为二阶低通滤波器的幅频特性曲线。

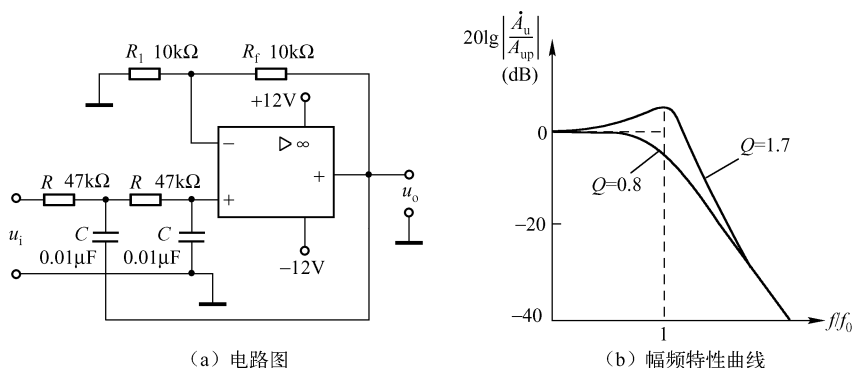


图 2-6 二阶低通滤波器

电路性能参数如下。

$A_{up} = 1 + \frac{R_f}{R_1}$ : 二阶低通滤波器的通带增益。

$f_0 = \frac{1}{2\pi RC}$ : 截止频率, 它是二阶低通滤波器通带与阻带的界限频率。

$Q = \frac{1}{3 - A_{up}}$ : 品质因数, 它的大小影响低通滤波器在截止频率处幅频特性的形状。

**2) 高通滤波器 (HPF)** 与低通滤波器相反, 高通滤波器用来通过高频信号, 衰减或抑制低频信号。

只要将图 2-6 所示低通滤波电路中起滤波作用的电阻、电容互换,即可变成二阶有源高通滤波器,如图 2-7 (a) 所示。高通滤波器的性能与低通滤波器相反,其频率响应和低通滤波器是“镜像”关系,仿照 LPH 的分析方法,不难求得 HPF 的幅频特性。

电路性能参数  $A_{up}$ 、 $f_0$ 、 $Q$  各量的含义同二阶低通滤波器。

图 2-7 (b) 为二阶高通滤波器的幅频特性曲线,可见,它与二阶低通滤波器的幅频特性曲线有“镜像”关系。

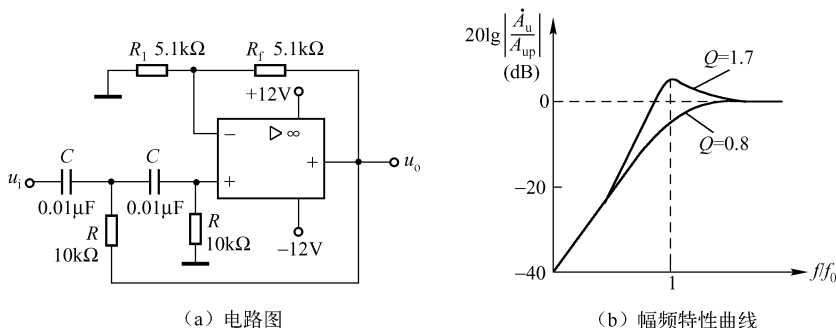


图 2-7 二阶高通滤波器

**3) 带通滤波器 (BPF)** 这种滤波器的作用是只允许在某一个通频带范围内的信号通过,而比通频带下限频率低和比上限频率高的信号均被衰减或抑制。

典型的带通滤波器可以从二阶低通滤波器中将其中一级改成高通而形成,如图 2-8 (a) 所示。电路性能参数如下。

通带增益:

$$A_{up} = \frac{R_4 + R_f}{R_4 R_1 C B}$$

中心频率:

$$f_0 = \frac{1}{2\pi\sqrt{R_2 C^2 \left( \frac{1}{R_1} + \frac{1}{R_3} \right)}}$$

通带宽度:

$$B = \frac{1}{C} \left( \frac{1}{R_1} + \frac{2}{R_2} - \frac{R_f}{R_3 R_4} \right)$$

选择性:

$$Q = \frac{\omega_0}{B}$$

图 2-8 (b) 为二阶带通滤波器的幅频特性曲线。该电路的优点是改变  $R_f$  和  $R_4$  的比例就可改变频宽而不影响中心频率。

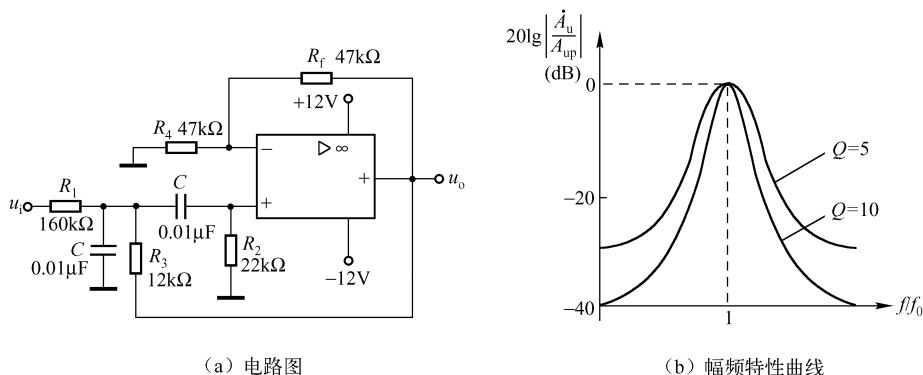


图 2-8 二阶带通滤波器



**4) 带阻滤波器 (BEF)** 如图 2-9 (a) 所示, 带阻滤波器电路的性能和带通滤波器相反, 即在规定的频带内, 信号不能通过 (或受到很大衰减或抑制), 而在其余频率范围, 信号则能顺利通过。

在双 T 网络后加一级同相比值运算电路就构成了基本的二阶有源 BEF。电路性能参数如下。

通带增益: 
$$A_{up} = 1 + \frac{R_f}{R_1}$$

中心频率: 
$$f_0 = \frac{1}{2\pi RC}$$

带阻宽度: 
$$B = 2(2 - A_{up})f_0$$

选择性: 
$$Q = \frac{1}{2(2 - A_{up})}$$

如图 2-9 (b) 所示为二阶带阻滤波器的幅频特性曲线。

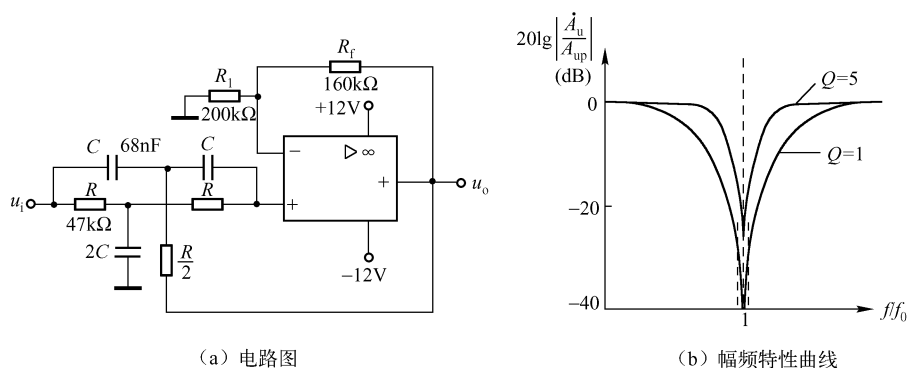


图 2-9 二阶带阻滤波器



## 2.3 组合逻辑电路

在数字电路中, 逻辑电路分为两大类: 组合逻辑电路和时序逻辑电路。本节介绍组合逻辑电路, 简称组合电路。

**组合逻辑电路的基本概念:** 任一时刻的输出状态只取决于该时刻各输入状态的组合, 与电路的原状态无关。电路只有从输入到输出的通路, 没有从输出到输入的反馈回路。电路只能由逻辑门构成, 不含记忆元件。

如图 2-10 所示, 在  $t=a$  时刻有输入  $X_1, X_2, \dots, X_n$ , 则在  $t=a$  时刻就有输出  $F_1, F_2, \dots, F_m$ , 每个输出都是输入  $X_1, X_2, \dots, X_n$  的函数。

$$\begin{cases} F_1 = f_1(X_1, X_2, \dots, X_n) \\ F_2 = f_2(X_1, X_2, \dots, X_n) \\ \vdots \\ F_m = f_m(X_1, X_2, \dots, X_n) \end{cases}$$

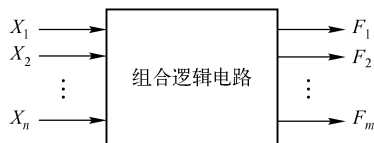


图 2-10 组合逻辑电路输入、输出关系图



下面将介绍组合逻辑电路中常用的电路，包括基本逻辑门电路、复合逻辑门电路、触发器、寄存器和计数器等。

### 1. 基本逻辑门电路

所谓“逻辑”是指事件的前因后果所遵循的规律，反映事物逻辑关系的变量称为逻辑变量。如果把数字电路的输入信号看做“条件”，把输出信号看做“结果”，则数字电路的输入与输出信号之间存在着一定的因果关系，即存在逻辑关系，能实现一定逻辑功能的电路称为逻辑门电路。

在数字电路中，逻辑门电路是实现某种逻辑关系的最基本的单元，任何复杂的组合电路及时序电路都可用逻辑门电路通过适当的组合连接而成。基本的逻辑门电路有与门、或门和非门。

**1) 与逻辑门** 如图 2-11 (a) 所示，开关 A 与 B 串联在回路中，两个开关都闭合时，信号灯点亮；若其中任意一个开关断开，信号灯就不会亮。这里开关 A、B 的闭合与信号灯亮的关系称为逻辑与，也称为逻辑乘。因此与逻辑可概括为：只有当决定一件事情的条件全部具备之后，这件事情才会发生。把这种因果关系称为与逻辑。

(1) 可以用列表的方式表示上述逻辑关系，称为真值表，如图 2-11 (b) 所示。

(2) 如果用二值逻辑 0 和 1 来表示，并设 1 表示开关闭合或灯亮，0 表示开关不闭合或灯不亮，则得到如图 2-11 (c) 所示的表格，称为逻辑真值表。

(3) 若用逻辑表达式来描述，则可写为  $L = A \cdot B$ 。

与逻辑门运算的规则为：“输入有 0，输出为 0；输入全 1，输出为 1。”

(4) 在数字电路中能实现与运算的电路称为与逻辑门电路，其逻辑符号如图 2-11 (d) 所示。

与逻辑门运算可以推广到多变量： $L = A \cdot B \cdot C \cdots$ 。

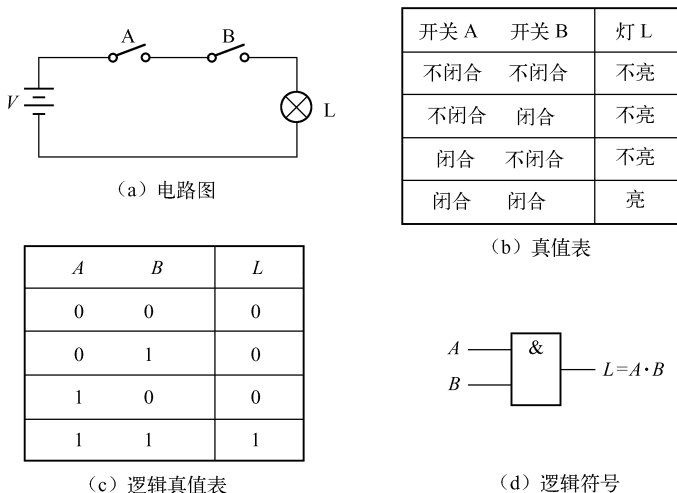


图 2-11 与逻辑门

**2) 或逻辑门** 如果把图 2-11 (a) 中的开关 A、B 改为并联再和电灯连接起来，开关控制电路如图 2-12 (a) 所示，显然，灯亮的条件是：只要开关 A 或 B 至少有一个闭合就

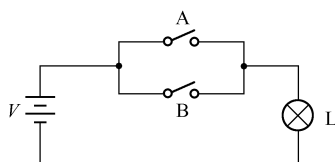
行。所以这种灯亮与开关闭合的关系是“或”逻辑。因此或逻辑可概括为：在决定一个事件发生的几个条件中，只要其中一个或一个以上的条件成立，事件就会发生，这种电路关系称为或逻辑关系。或逻辑可表示为  $L = A + B$ ，式中“+”为或逻辑运算符号， $A + B$  读成“A 或 B”。

或逻辑门运算的真值表如图 2-12 (b) 所示，逻辑真值表如图 2-12 (c) 所示。若用逻辑表达式来描述，则可写为  $L = A + B$ 。

或逻辑门运算的规则为：“输入有 1，输出为 1；输入全 0，输出为 0。”

在数字电路中能够实现或运算的电路称为或逻辑门电路，其逻辑符号如图 2-12 (d) 所示。

或运算也可以推广到多变量： $L = A + B + C + \dots$ 。



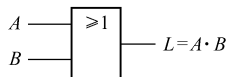
(a) 电路图

开关 A	开关 B	灯 L
不闭合	不闭合	不亮
不闭合	闭合	亮
闭合	不闭合	亮
闭合	闭合	亮

(b) 真值表

A	B	$L = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(c) 逻辑真值表



(d) 逻辑符号

图 2-12 或逻辑门

**3) 非逻辑门运算** 非运算：某事情发生与否，仅取决于一个条件，而且是对该条件的否定。即条件具备时事情不发生；条件不具备时事情才发生。

如图 2-13 (a) 所示的电路，当开关 A 闭合时，灯不亮；而当 A 不闭合时，灯亮。其真值表如图 2-13 (b) 所示，逻辑真值表如图 2-13 (c) 所示。若用逻辑表达式来描述，则可写为  $L = \bar{A}$ 。非逻辑门运算的规则为： $\bar{0} = 1$ ； $\bar{1} = 0$ 。

在数字电路中实现非运算的电路称为非逻辑门电路，其逻辑符号如图 2-13 (d) 所示。

## 2. 复合门电路

在实际使用中，可以将上述的基本逻辑门电路组合起来，构成常用的复合逻辑门电路，以实现各种逻辑功能。如将与门、或门、非门经过简单组合，可构成另一些复合逻辑门。常用的复合逻辑门有“与非门”、“或非门”、“异或门”等。

**1) 与非逻辑门** 与非逻辑门是由与运算和非运算组合而成的，如图 2-14 所示。

**2) 或非逻辑门** 或非逻辑门是由或运算和非运算组合而成的，如图 2-15 所示。

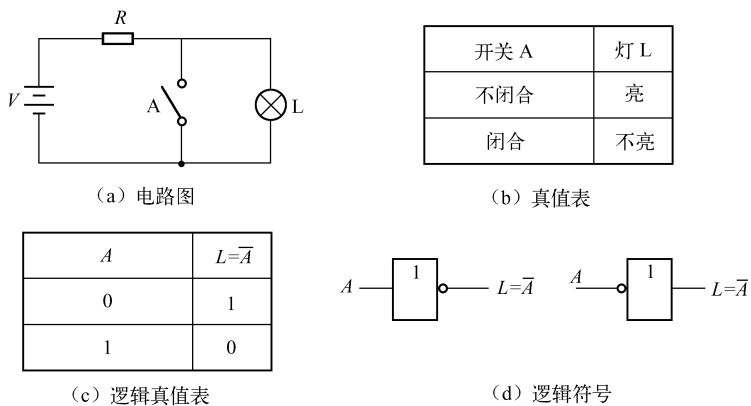


图 2-13 非逻辑门

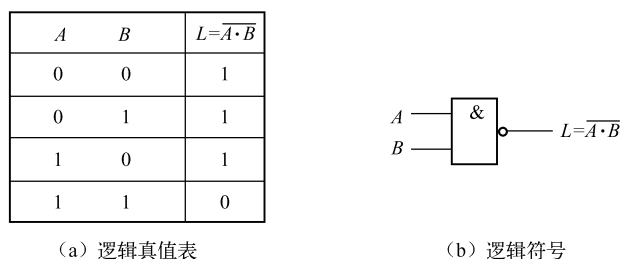


图 2-14 与非逻辑门

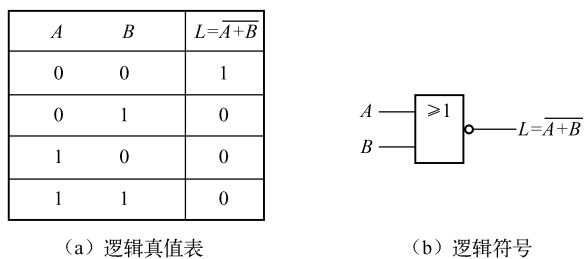


图 2-15 或非逻辑门

**3) 异或逻辑门** 异或门是一种二变量逻辑运算，当两个变量取值相同时，逻辑函数值为 0；当两个变量取值不同时，逻辑函数值为 1。异或门的逻辑真值表和相应的逻辑符号如图 2-16 所示。

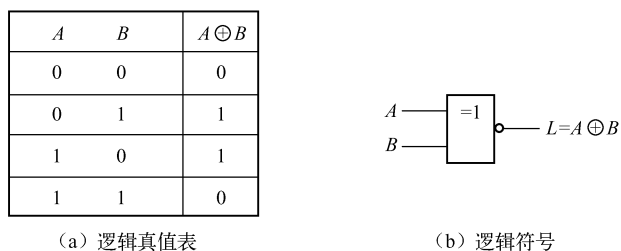


图 2-16 异或逻辑运算



## 2.4 触发器

触发器是构成时序电路中的基本单元,触发器能够存储1位二进制码的逻辑电路,它有两个互补输出端,其输出状态不仅与输入有关,而且还与原先的输出状态有关。触发器具有不同的逻辑功能,在电路结构和触发方式方面也有不同的种类。在分析触发器的功能时,一般可用功能表、特性方程和状态图来描述其逻辑功能。研究触发方式时,主要是分析其输入信号的加入与触发脉冲之间的时间关系。

触发器是双稳态触发器的简称。双稳态触发器由于能够保持两种不同的稳定输出状态,并且在一定的条件下这两种稳定输出状态可以相互转化而得名。

### 1. 触发器的电路结构与工作原理

触发器具有数码的记忆功能,能够实现的逻辑功能有:计数功能;置数功能——置1、置0功能;保持功能。

(1) 按结构不同触发器可以分为置位、复位触发器(基本RS触发器、同步RS触发器),主从型触发器(由主触发器和从触发器构成),边沿型触发器(上升沿触发、下降沿触发和利用传输延时的边沿触发器)。

(2) 按功能不同可将触发器分为RS型、JK型、D型和T型(含T'型)触发器。

(3) 按器件分有TTL型和CMOS型触发器。

(4) 按照逻辑功能的不同特点可分为RS触发器、JK触发器、D触发器和T触发器等几种类型。

触发器输出端状态和输入激励信号之间的关系称为触发器的逻辑功能。描述触发器的功能通常有5种不同的方式,或者称为5种功能描述方法。

(1) 特性表(功能表):以表格形式描述触发器的逻辑功能,输入(激励)和输出可用逻辑值0或1表示,也可用逻辑电平高(H)或低(L)表示。

(2) 特性方程(特征方程):以表达式形式描述触发器的逻辑功能,但特征方程的两边表示的时间是不一致的。

(3) 状态转换图(状态图):以图形形式描述触发器状态转换的激励条件。

(4) 激励表:以表格形式描述触发器状态转换的激励条件。

(5) 时序图:以时序波形形式描述触发器在相应激励下状态转换的过程。

这些描述方法与组合逻辑电路的逻辑功能描述方法是相似的,根据时序电路的特殊情况而有所不同。比如描述门电路或组合逻辑电路输入、输出关系的是电路的逻辑函数,在这里称为触发器的特性方程,以便和逻辑函数相区别;真值表在这里称为特性表或功能表;状态转换图反映的是时序电路的状态转换规律及相应输入、输出取值关系的图形;激励表的功能与状态图是一致的,只是表示形式不同;时序图是时序电路的工作波形,能直观地描述时序电路的输入信号、时钟信号、输出信号及电路的状态转换在时间上的对应关系。

需要特别强调,这些描述方法所表示的输出状态与激励信号之间的关系中还必须有时钟信号的配合(基本RS触发器除外),没有有效的时钟信号到来,即使加上了激励信号,触发器的状态也不会改变,即维持原来的 $Q$ 状态不变。除了时序波形中画时钟信号外,其他

描述方法中一般不给出与时钟信号的关系。

## 2. 基本（双稳态）RS 触发器

下面介绍 RS 触发器的基本电路结构和工作原理。

由两个首尾交叉连接的与非门构成的基本 RS 触发器，如图 2-17（a）所示，图 2-17（b）是基本 RS 触发器的逻辑符号。

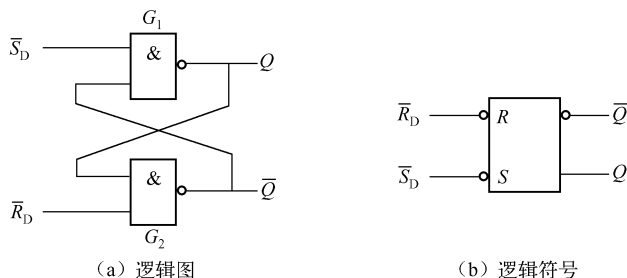


图 2-17 两个与非门构成的基本 RS 触发器

规定将触发器的  $Q$  端状态称为触发器的状态。 $Q=0$ ， $\bar{Q}=1$  称触发器处于 0 状态； $Q=1$ ， $\bar{Q}=0$  则称触发器处于 1 状态。所以触发器具有两个稳定状态：1 态和 0 态，因而又将其称为“双稳态”电路。它是实现记忆的基础，可分别用来表示二进制数 0 和 1，以及两个相互对立的各逻辑状态。当然，在每一具体时刻，触发器只能处于两个稳态中的一个，即为 0 或 1。

触发器具有置数功能（预置功能）。触发器的状态可以通过输入信号置为 1（置位）或置为 0（复位），而输入信号撤销后，其状态仍然保持。在电路中置位端常以  $S$ （SET）表示，复位端常以  $R$ （RESET）表示，预置数功能又称为直接置数（置位），通常对信号  $S$  和  $R$  加下标“ $D$ ”写成  $S_D$ 、 $R_D$ ，若置数信号是低电平有效则写成  $\bar{S}_D$ 、 $\bar{R}_D$ 。在外加信号的作用下，可能使触发器状态发生变化，即由一个稳态（0 或 1）变为另一个稳态（1 或 0），这种触发器状态的转换称为翻转，引起翻转的输入信号称为触发信号。一般来说，触发信号一旦使触发器状态翻转，就可以撤销，而由此引起的触发器状态则维持下来。与置数作用不同的是，触发信号一般要与时钟脉冲 CP 配合才能有效。

RS 触发器有两个输入  $\bar{R}_D$ 、 $\bar{S}_D$  和两个输出  $Q$ 、 $\bar{Q}$ ，符号方框内的  $R$ 、 $S$  分别表示复位和置位输入。复位端（ $R$ ）加低电平，即  $\bar{R}_D=0$ ，与非门  $G_2$  见“0”输出“1”；置位端（ $S$ ）加低电平，即  $\bar{S}_D=0$ ，与非门  $G_1$  见“0”输出“1”。因此，复位和置位输入信号中只能有一个为低电平，低电平加在复位端，触发器复位， $Q=0$ ；低电平加在置位端，触发器置位， $Q=1$ 。所以，RS 触发器称为置位复位触发器。在逻辑符号中用“ $\circ$ ”表示低电平是有效信号。

表 2-3 用两个与非门构成的 RS 触发器的功能表

$\bar{R}_D$	$\bar{S}_D$	$Q$
1	0	1
0	1	0
1	1	不变
0	0	不定

通过在  $\bar{R}_D$ 、 $\bar{S}_D$  端加不同的信号同时考虑  $Q$  和  $\bar{Q}$  的状态，可得其功能表如表 2-3 所示。当  $\bar{R}_D = \bar{S}_D = 0$  时，两个输出端  $Q$  和  $\bar{Q}$  都为 1，破坏了触发器输出互补的逻辑关系，特别当

$\bar{R}_D = \bar{S}_D = 0$  信号同时消失时, 由于  $G_1$ 、 $G_2$  门传输延迟时间的不确定, 触发器的新状态也是不确定的, 故在真值表中描述为不定状态, 这种输入情况应避免使用。

### 3. 同步 (钟控) RS 触发器

同步 RS 触发器是由一个时钟脉冲信号 CP 控制的 RS 触发器。当要求触发器状态不是单纯地受  $R$ 、 $S$  端信号控制, 还要求按一定时间节拍把  $R$ 、 $S$  端的状态反映到输出端时, 就必须再增加一个控制端, 只有控制端出现脉冲信号时, 触发器才动作, 至于触发器输出变到什么状态, 仍然由  $R$ 、 $S$  端的高低电平来决定, 采用这种触发方式的触发器, 称为同步 (钟控) RS 触发器, 如图 2-18 所示, 其中图 (a) 为由与或非门构成, 图 (b) 为由与非门构成。

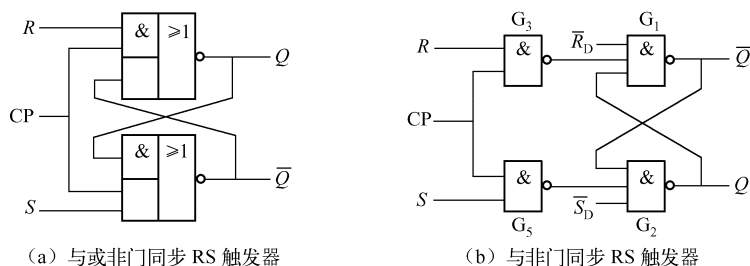


图 2-18 同步 RS 触发器

分析图 2-18 (b), 其中  $G_1$ 、 $G_2$  门构成基本 RS 触发器,  $G_3$ 、 $G_5$  门组成控制电路, CP 是控制脉冲。所谓同步就是触发器状态的改变与时钟脉冲同步。当  $CP = 0$  时,  $G_3$ 、 $G_5$  门被封锁,  $R$ 、 $S$  状态不能进入,  $G_3$ 、 $G_5$  门输出均为高电平, 触发器输出保持原来状态; 当  $CP = 1$  时,  $R$ 、 $S$  信号才能经过  $G_3$ 、 $G_5$  门影响到输出。 $\bar{S}_D$  为直接置 1 端,  $\bar{R}_D$  为直接置 0 端, 它们的电平可以不受 CP 信号的控制而直接影响触发器的输出。利用基本 RS 触发器的真值表, 可得同步 RS 触发器的功能表如表 2-4 所示。因为有 CP 脉冲的加入, 要考虑 CP 脉冲作用前后  $Q$  端的状态, 所以将 CP 脉冲作用前  $Q$  端的状态用  $Q^n$  表示, 称为触发器的原状态, CP 脉冲作用后  $Q$  端的状态用  $Q^{n+1}$  表示, 称为触发器的次状态。将这种考虑了 CP 脉冲作用前后  $Q$  端状态的表格称为特性表或功能表。

同步 RS 触发器的逻辑符号如图 2-19 (a) 和 (b) 所示, 图 2-19 (a) 为惯用符号, 图 2-19 (b) 为标准符号。在标准图形符号中, 为了表示时钟输入对激励输入 ( $R$ 、 $S$ ) 的这种控制作用, 时钟端用控制字符 C 加标记序号 1 表示, 置位端 S 前加标记序号 1 写成 1S, 同理, 复位端写成 1R, 表示它/它们是受 C1 控制的置位、复位端。图 2-19 (b) 中 R 和 S 的前面没有标记序号, 表示是不受时钟控制的置位、复位端, 也称为异步置位端和异步复位端。

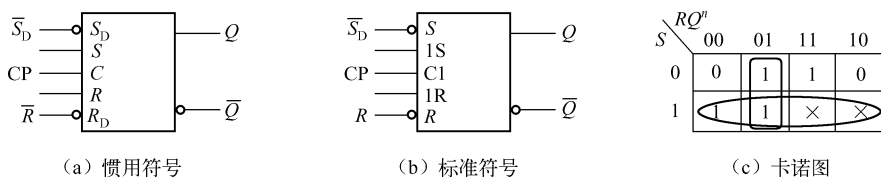


图 2-19 同步 RS 触发器的逻辑符号和卡诺图

表 2-4 同步 RS 触发器功能表

S	R	$Q^n$	$Q^{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	不定
1	1	1	不定

根据表 2-4 和如图 2-19 (c) 所示的卡诺图, 可得到同步 RS 触发器的特征方程如下:

$$Q^{n+1} = S + \bar{R}Q^n \quad (2-5)$$

$$SR = 0 \quad (\text{约束条件})$$

特征 (特性) 方程同样描述了 RS 触发器的逻辑功能。将 RS 的不同状态代入特征方程可得: RS = 00,  $Q^{n+1} = Q^n$ , 触发器状态不变; RS = 01,  $Q^{n+1} = 1$ , 触发器置位; RS = 10,  $Q^{n+1} = 0$ , 触发器复位; RS = 11 不满足约束条件, 这是一种禁止输入状态。由图 2-18 可见, CP = 1 时输出端不互补, 既不是 0 状态, 也不是 1 状态; CP 变为 0 时, 钟控门 ( $G_3$ 、 $G_5$ ) 输出同时变为 1, 触发器状态不确定。根据分析, 可得如表 2-5 所示的 RS 触发器简化功能表。比较表 2-3 和表 2-5 可以发现, 用  $\bar{R}$  替代  $R_D$ ,  $\bar{S}$  替代  $S_D$ ,  $Q^{n+1}$  替代  $Q$ , 它们实质上是相同的。

#### 4. 主从 RS 触发器

同步 RS 触发器在 CP = 1 期间接收 R、S 信号, 若 CP = 1 期间 R、S 信号发生变化, 则 Q 端状态会发生多次翻转, 这种现象称为空翻, 在某些场合会造成逻辑混乱。为克服空翻现象, 引入主从结构的触发器。图 2-20 (a) 所示为主从 RS 触发器逻辑电路图, 图 2-20 (b) 为逻辑符号。

表 2-5 RS 触发器简化功能表

R	S	$Q^{n+1}$
0	0	$Q^n$
0	1	1
1	0	0
1	1	×

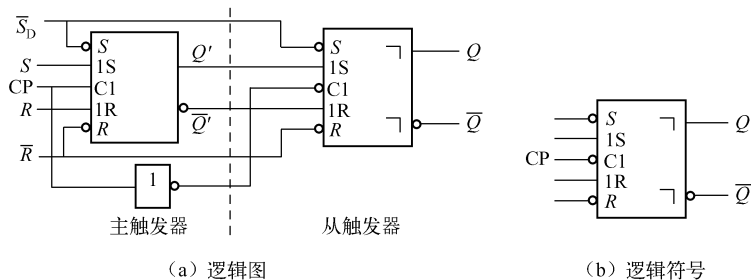


图 2-20 由两个同步 RS 触发器组成的主从 RS 触发器

由图 2-20 可见主从 RS 触发器分别由两个互补的时钟脉冲信号控制两个同步 RS 触发器, 在 CP = 0 时, 主触发器被封锁、从触发器被打开, 主触发器的状态决定从触发器的状态。由于 CP = 0, 主触发器被封锁, 所以 R、S 信号变化不能直接影响到输出。在 CP = 1 时, 主触发器打开、从触发器被封锁, Q 维持不变, R、S 信号决定主触发器的状态。因此无论 CP 为高还是低, 主、从触发器总是一个打开, 另一个被封锁, R、S 状态的改变不可能直接影响输出状态, 从而解决了空翻现象。

主从 RS 触发器的功能与同步 RS 触发器完全一样, 都是在 CP 的作用下将 R、S 端的状态反映给输出端。同步 RS 触发器的翻转发生在 CP 脉冲的上升沿。主从 RS 触发器由两个同步触发器组成, CP 上升沿时, 主触发器翻转, 从触发器封锁, Q 不变; CP 下降沿时, 主触发器封锁, 而从触发器打开, 将主触发器的状态反映到 Q 端。所以主从触发器翻转发生在

CP 脉冲的下降沿, 逻辑符号中时钟端 C1 的小圈表示了这层含义, 输出端的符号 “ $\neg$ ” 表示主从触发输出。

凡是在时钟信号作用下逻辑功能满足表 2-4 特性表所规定的逻辑功能的电路, 就是 RS 触发器。RS 触发器的次状态与激励  $R$ 、 $S$  有关, 也与其原状态有关, 将表 2-4 表示的触发器状态转换关系用图形表示, 得到图 2-21 所示的状态转换图。

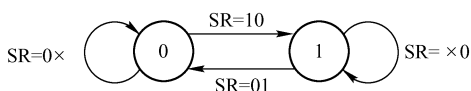


图 2-21 RS 触发器符号和状态图

从结构上分, RS 触发器可分为基本 RS 触发器、同步 RS 触发器和主从 RS 触发器。无论什么结构, 表 2-4 和图 2-21 都反映了 RS 触发器的逻辑功能, 所不同的是, 不同结构的触发器状态变化的时间不一样。基本 RS 触发器的输出直接由  $R$ 、 $S$  状态决定, 原状态  $Q^n$  是  $R$ 、 $S$  变化前的触发器状态, 次状态  $Q^{n+1}$  是  $R$ 、 $S$  变化后的触发器状态, 即按输入  $R$ 、 $S$  信号变化划分原状态和次状态。同步 RS 触发器在外接时钟信号  $CP=1$  期间, 根据  $R$ 、 $S$  的状态改变。换句话说,  $CP=0$  期间, 无论输入  $R$ 、 $S$  如何变化, 触发器的状态也不会改变。主从 RS 触发器在时钟脉冲  $CP$  的下降沿 (或后沿) 触发器的状态变化, 但是, 是  $CP=1$  期间  $R$ 、 $S$  的状态决定  $CP$  下跳后触发器的状态。

## 5. 触发器的功能

### 1) JK 触发器

(1) 功能分析。由 RS 触发器特性表 (见表 2-4) 可知当  $R=S=1$  时, 触发器的输出状态不定, 须避免使用, 这给使用带来不便, 为此引入 JK 触发器从电路设计上避免这种情况出现。

考虑到 RS 触发器  $Q$  和  $\bar{Q}$  互补的特点, 将输出  $Q$  和  $\bar{Q}$  反馈到输入端, 通过两个与门使加到  $R$  和  $S$  端的信号不能同时为 1, 以满足 RS 触发器要求的约束条件。为区别于原来的 RS 触发器, 将对应于原图中的  $R$  用  $K$  表示、 $S$  用  $J$  表示, 如图 2-22 (a) 所示。这种改接后的电路称为主从 JK 触发器, 其逻辑符号如图 2-22 (b) 所示。

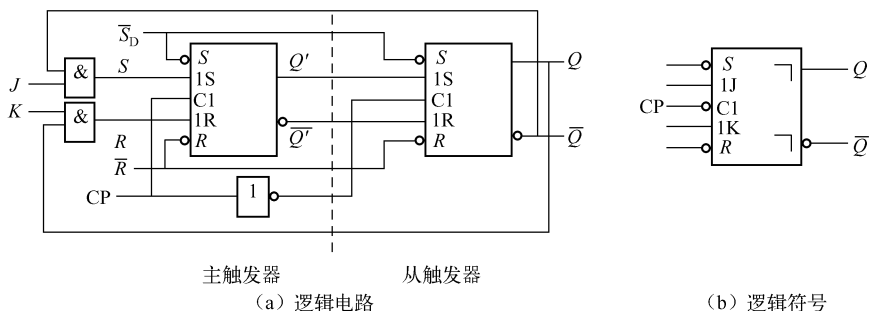


图 2-22 主从 JK 触发器

根据主从 RS 触发器的特性表 (见表 2-4), 通过比照即可获得主从 JK 触发器的特性表, 如表 2-6 和表 2-7 所示。在画 JK 触发器工作波形及分析 JK 触发器组成的时序逻辑电路时, 要求熟记其特性表或特征方程。



表 2-6 JK 触发器功能表

$J$	$K$	$Q^n$	$Q^{n+1}$	说 明
0	0	0	0	$Q^n$
0	0	1	1	$Q^n$
0	1	0	0	与 J 端状态相同
0	1	1	0	与 J 端状态相同
1	0	0	1	与 J 端状态相同
1	0	1	1	与 J 端状态相同
1	1	0	1	$\overline{Q^n}$
1	1	1	0	$\overline{Q^n}$

表 2-7 JK 触发器简化功能表

$J$	$K$	$Q^{n+1}$
0	0	$Q^n$
0	1	0
1	0	1
1	1	$\overline{Q^n}$

JK 触发器的状态转换图如图 2-23 (a) 所示, 图 2-23 (b) 给出了负边沿 JK 触发器的逻辑符号, 方框中的小三角表示边沿触发器。这也是一种在时钟下降沿才能改变状态的触发器, 但和主从 JK 触发器有所区别, 该触发器 CP 下降沿后的状态由 CP 下降沿前一刻的输入  $J$  和  $K$  决定。而主从 JK 触发器的次状态由  $CP=1$  期间的  $J$  和  $K$  确定。时钟前沿变化的边沿 JK 触发器逻辑符号中时钟端没有小圈。

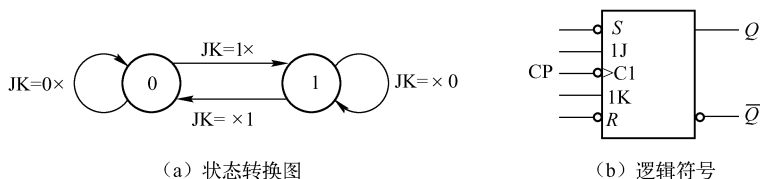


图 2-23 JK 触发器状态转换图和负边沿 JK 触发器逻辑符号

由表 2-6 所示功能表可得 JK 触发器的特征方程 (又称次态方程) 如下:

$$Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n \quad (2-6)$$

从结构上分, JK 触发器可分为主从型 JK 触发器和边沿型 JK 触发器。主从型 JK 触发器只能在 CP 脉冲的下降沿触发, 而边沿型 JK 触发器可以在 CP 脉冲的下降沿或上升沿触发。相对于边沿型触发器, 主从型 JK 触发器的抗干扰能力差些, 且存在一次变化问题。

(2) 主从 JK 触发器的一次变化问题。主从型触发器有两个重要的动作特点:

- 触发器状态的转换分两步进行, 第一步在  $CP=1$  期间主触发器接收输入激励信号, 第二步当 CP 下降沿到来时, 从触发器接收主触发器的激励进行转换。
- 在  $CP=1$  期间, 输入激励信号都将对主触发器起控制作用, 这就要求在  $CP=1$  期间输入的激励信号不能发生突变, 否则就不能再用通常给出的动作特性规律来决定触发器的状态。输入激励信号在  $CP=1$  期间的变化统称为干扰, 对于这种干扰必须考虑在  $CP=1$  期间输入激励信号的整个变化过程后才能确定触发器状态如何转换。

就主从型 JK 触发器而言, 经分析可按下述的方法处理这一类干扰:

- 在  $CP=1$  期间, 若输入激励信号出现负向干扰, 则这一干扰对触发器状态转换不起作用。也就是说, 当 CP 信号下降沿到来时, 触发器状态转换由负向干扰之前的输入激励信号决定。
- 在  $CP=1$  期间, 若输入激励信号中  $J$  信号上出现正向干扰, 又若此时触发器状态处于 0 态, 则这一干扰将起激励作用。也就是说, 当 CP 信号下降沿到来时, 触发器状态转换应取  $J=1$  的关系再视  $K$  值决定。
- 在  $CP=1$  期间, 若输入激励信号中  $K$  信号上出现正向干扰, 又若此时触发器状态处于 1 态, 则这一

干扰也将起激励作用。也就是说,当 CP 信号下降沿到来时,触发器状态转换应取  $K=1$  的关系再视  $J$  值决定。

**2) D 触发器** JK 触发器功能较完善,应用广泛,但需两个输入控制信号 ( $J$  和  $K$ )。如果在 JK 触发器的  $K$  端前面加上一个非门再接到  $J$  端,如图 2-24 所示,使输入端只有一个,在某些场合用这种电路进行逻辑设计可使电路得到简化,将这种触发器的输入端符号改用  $D$  表示,称为 D 触发器。

由 JK 触发器的功能表可得 D 触发器的功能表如表 2-8 所示。

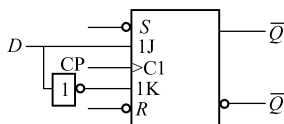


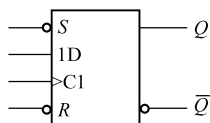
图 2-24 D 触发器构成

表 2-8 D 触发器的功能表

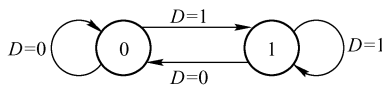
$D$	$Q^{n+1}$
0	0
1	1

D 触发器的逻辑符号和状态转换图如图 2-25 所示,图中 CP 输入端处无小圈,表示在 CP 脉冲上升沿触发。除了异步置 0 置 1 端  $R$ 、 $S$  外,只有一个控制输入端  $D$ 。因此 D 触发器的特性表比 JK 触发器的特性表简单。D 触发器的特征方程为:

$$Q^{n+1} = D$$



(a) 逻辑符号

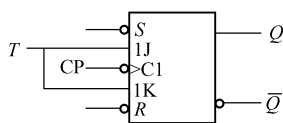


(b) 状态转换图

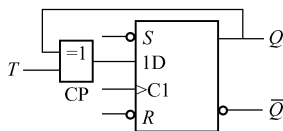
图 2-25 D 触发器

D 触发器的抗干扰能力强,工作时,对 CP 脉冲宽度的要求没有主从 JK 触发器那么苛刻。

**3) T 触发器** T 触发器又称受控翻转型触发器。这种触发器的特点很明显:  $T=0$  时,触发器由 CP 脉冲触发后,状态保持不变;  $T=1$  时,每来一个 CP 脉冲,触发器状态就改变一次。T 触发器并没有独立的产品,而是由 JK 触发器或 D 触发器转换而来,如图 2-26 所示。



(a) JK 触发器构成 T 触发器



(b) D 触发器构成 T 触发器

图 2-26 T 触发器

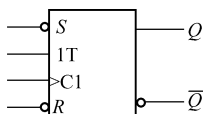
T 触发器的功能如表 2-9 所示。由功能表可写出 T 触发器的特征方程为:

$$Q^{n+1} = T \overline{Q^n} + \overline{T} Q^n = T \oplus Q^n \quad (2-7)$$

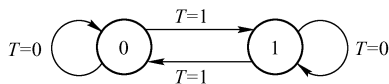
T 触发器的逻辑符号和状态转换图如图 2-27 所示。

表 2-9 T 触发器功能表

$T$	$Q^{n+1}$
0	$Q^n$
1	$\overline{Q^n}$



(a) 逻辑符号



(b) 状态转换图

图 2-27 T 触发器

4) **T'触发器** T'触发器又称为翻转型(计数型)触发器,其功能是在脉冲输入端每收到一个 CP 脉冲,触发器输出状态就改变一次。T'触发器也没有独立的产品,主要由 JK 触发器和 D 触发器转换而来,令  $J = K = 1$  或  $D = \overline{Q^n}$ ,如图 2-28 所示。所以可得其特征方程为:

$$Q^{n+1} = \overline{Q^n} \quad (2-8)$$

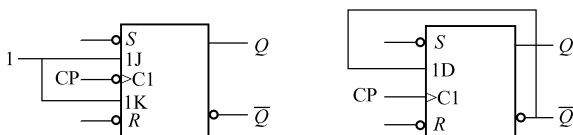


图 2-28 计数型触发器

不同类型的触发器可以通过附加组合电路实现不同逻辑功能之间的转换,如上述 JK 和 D 触发器转换成 T 和 T' 触发器。同样,JK 触发器和 D 触发器之间也能实现相互转换,转换后的触发器结构不变,主从型仍然是主从型,边沿型仍然是边沿型。



## 2.5 思考与练习

1. 如图 2-29 所示电路中,稳压管稳定电压  $U_Z = 6V$ ,电阻  $R_1 = 10k\Omega$ ,电位器  $R_f = 10k\Omega$ ,试求调节  $R_f$  时输出电压  $u_o$  的变化范围,并说明改变电阻  $R_L$  对  $u_o$  有无影响。

2. 如图 2-30 所示为一电压-电流变换电路,试求输出电流  $i_o$  与输入电压  $u_i$  的关系,并说明改变负载电阻  $R_L$  对  $i_o$  有无影响。

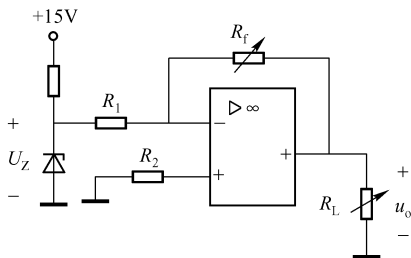


图 2-29 习题 2-1 图

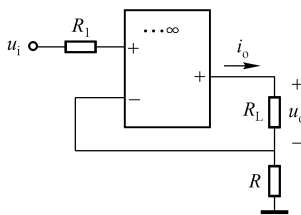


图 2-30 习题 2-2 图

3. 求如图 2-31 所示电路中  $u_o$  与  $u_i$  的关系。

4. 电路及  $u_{i1}$ 、 $u_{i2}$  的波形如图 2-32 所示,试画出对应  $u_o$  的波形。

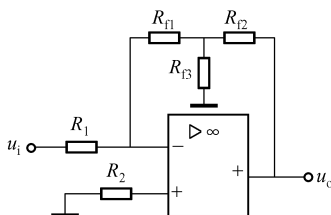


图 2-31 习题 2-3 图

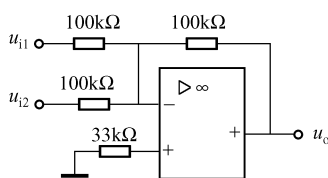
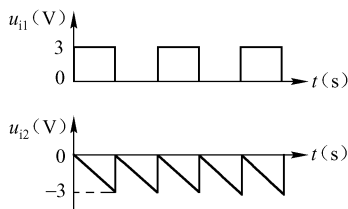


图 2-32 习题 2-4 图



5. 已知利用运放实现运算： $u_o = 2(u_{i1} + u_{i2}) - 0.5u_{i3}$ ，求完成上述功能的电路 ( $R_F = 20\text{k}\Omega$ )。

6. 无源滤波电路和有源滤波电路各有什么特点？各适用于什么场合？如何识别滤波电路的类型？

7. 简述滤波器的功能、分类及主要特性参数。

8. 分析如图 2-33 所示的同步时序逻辑电路：(1) 写出驱动方程和状态方程；(2) 确定电路的逻辑功能。

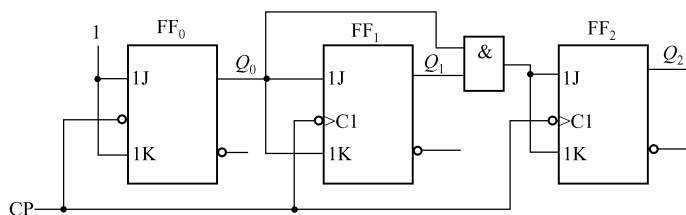


图 2-33 习题 2-8 图

9. 一个二阶带通滤波器电路如图 2-34 所示，其中  $R_1 = 56\text{k}\Omega$ ， $R_2 = 2.7\text{k}\Omega$ ， $R_3 = 4.7\text{k}\Omega$ ， $R_0 = 20\text{k}\Omega$ ， $R = 3.3\text{k}\Omega$ ， $C_1 = 1\mu\text{F}$ ， $C_2 = 0.1\mu\text{F}$ 。求电路品质因数  $Q$  与通带中心频率  $f_0$ 。当外界条件使电容  $C_2$  增大或减小 1% 时， $Q$  与  $f_0$  变为多少？当电阻  $R_2$  增大或减小 1%，或当电阻  $R_2$  减小 5% 时， $Q$  与  $f_0$  变为多少？

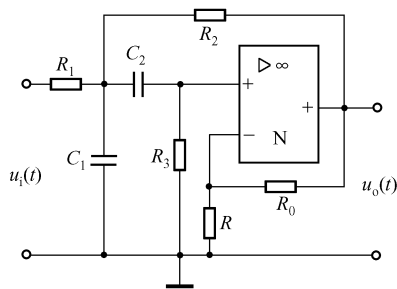


图 2-34 习题 2-9 图

10. 在一个射击游戏中，射手可打三枪，一枪打鸟，一枪打鸡，一枪打兔子，规则是命中不少于两枪者获奖。试用与非门设计一个判别得奖的电路。

11. 试用 JK 触发器设计一个序列检测器，当连续输入 1101 时，该电路输出 1，否则输出 0。

12. 分析如图 2-35 所示的异步时序逻辑电路。

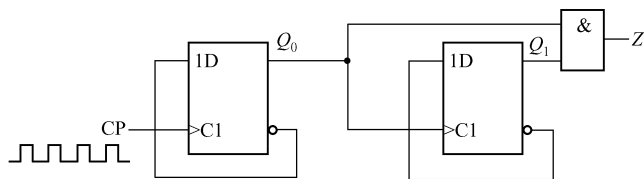


图 2-35 习题 2-12 图

13. 用边沿 D 触发器和门电路设计一个二进制数码串行加法器。
14. 编码器的逻辑功能是什么？优先编码器与一般编码器有何区别？
15. 要区别 24 个不同信号，或者说给 24 个输入信号编码，需要几位二进制代码？电路有多少个输出？如果区别 64 个信号又将如何？
16. 什么叫译码器？有哪些常用译码器？各有何特点？
17. 某工厂有三个车间和一个自备电站，电站有 X、Y 两台发电机，当一个车间开工时，只需启动 X，任意两个车间开工时，只需启动 Y，三个车间同时开工时，X、Y 都应启动，试设计一个启动发电机 X、Y 的控制电路，并要求逻辑门个数最少。
18. 有一个火灾报警系统，设有烟感、温感和紫外光感三种不同类型的火灾探测器，为了防止误报警，只有当其中两种或两种以上探测器发出探测信号时，报警系统才产生报警信号。试用与非门设计该报警信号电路，要求写全分析步骤。

## 第3章 51 单片机的结构与原理

MCS-51 系列单片机是 Intel 公司在 MCS-48 系列的基础上，在 20 世纪 80 年代初推出的高性能的 8 位单片机。它与 48 系列相比，在片内存储器容量、I/O 口的功能及指令系统功能等方面都大大得到加强。MCS-51 系列单片机特别适用于实时控制、智能仪表、主从结构的多机系统等领域，是工业检测、控制领域中理想的 8 位单片机。

从应用的角度看，MCS-51 单片机具有如下一些特点：

- 集成度高；
- 系统结构简单；
- 系统扩展方便；
- 可靠性高；
- 处理功能强、速度快；
- 容易产品化。



### 3.1 51 单片机的硬件结构

MCS-51 单片机的结构特点如下：

- 一个 8 位微处理器 CPU；
- 数据存储器 RAM 和特殊功能寄存器 SFR；
- 内部程序存储器 ROM；
- 两个定时/计数器，用以对外部事件进行计数，也可用做定时器；
- 4 个 8 位可编程的 I/O（输入/输出）并行端口，每个端口既可做输入，也可做输出；
- 一个串行端口，用于数据的串行通信；
- 中断控制系统；
- 内部时钟电路。

#### 1. 51 单片机的基本结构

MCS-51 单片机和微型机一样，是由运算器、控制器、存储器、输入设备、输出设备 5 个基本部分组成的，如图 3-1 所示。

1) CPU CPU 是单片机的核心部分，CPU 包括两个基本部分：运算器和控制器。

(1) 运算器。运算器即算术逻辑运算单元 ALU (Arithmetic Logic Unit)，是进行算术或逻辑运算的部件，可实现算术运算和逻辑运算。操作的结果一般送回累加器 ACC (Accumulator)，而其状态信息送至程序状态寄存器 PSW (Program Status Word)。

(2) 控制器。控制器是用来控制计算机工作的部件。控制器接收来自存储器的指令，使各部分协调工作，完成指令所规定的操作。

2) 内部数据存储器 MCS-51 单片机共有 256B（地址为 00H ~ FFH）的数据存储器，其中高 128B（地址为 80H ~ FFH）被专用寄存器占用，能作为寄存器供用户使用的只是低

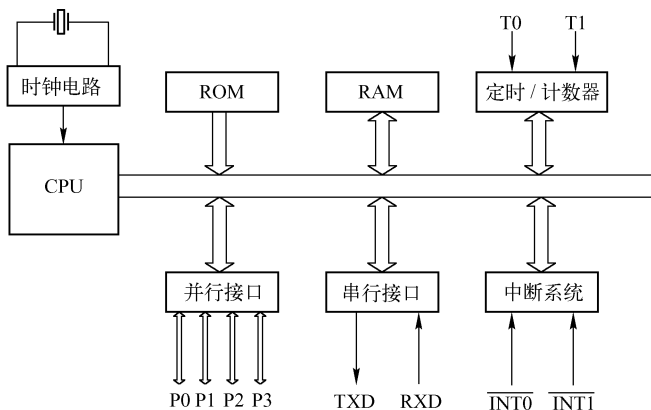


图 3-1 MCS-51 单片机的基本结构

128B（地址为 00H ~ 7FH），用于存放可读写的数，如程序执行过程中的变量。

**3) 内部程序存储器** MCS-51 单片机内部程序存储器的空间范围为 0 ~ 64KB（地址为 0000H ~ 0FFFH），用于存放程序、原始数据或表格常数。

**4) 定时/计数器** MCS-51 单片机有两个 16 位的定时/计数器，每个定时/计数器都可以设置成计数方式，用于对外部事件进行计数；也可以设置成定时方式，并可以根据计数或定时的结果实现对单片机运行的控制。

**5) 并行 I/O 口** MCS-51 单片机一般有 32 根 I/O 口线（某些型号的 MCS-51 单片机），分别属于 4 个 8 位并行 I/O 口 P0、P1、P2 和 P3。每个口都可以用做输入和输出。其中 P0 和 P2 口在存储器扩展中又可以作为地址和数据总线使用，P3 口又是一个双工能口，在应用中以第二功能为主。虽然各个口的功能有所不同，但其结构和工作过程基本相似。

1	P1.0	VCC	40
2	P1.1	P0.0	39
3	P1.2	P0.1	38
4	P1.3	P0.2	37
5	P1.4	P0.3	36
6	P1.5	P0.4	35
7	P1.6	P0.5	34
8	P1.7	P0.6	33
9	RST/VPD	P0.7	32
10	RXD P3.0	EA/VPP	31
11	TXD P3.1	ALE/PROG	30
12	INT0 P3.2	PSEN	29
13	INT1 P3.3	P2.7	28
14	T0 P3.4	P2.6	27
15	T1 P3.5	P2.5	26
16	WR P3.6	P2.4	25
17	RD P3.7	P2.3	24
18	XTAL2	P2.2	23
19	XTAL1	P2.1	22
20	VSS	P2.0	21

图 3-2 MCS-51 单片机的引脚

## 2. 51 单片机的引脚描述

**1) 信号引脚介绍** MCS-51 单片机采用 40 脚双列直插式的 DIP40 封装，还提供较小尺寸表面封装形式的 PQFP/TQFP44，引脚排列如图 3-2 所示。40 个引脚大致可分为 4 类：I/O、控制、时钟和电源引脚。

(1) I/O 口（32 条）。

➤ P0 口（P0.0 ~ P0.7），P0.0 是最低位，有两种功能。

☆ 通用 I/O 接口：无片外存储器时，P0 口可作通用 I/O 接口使用。

☆ 地址/数据口。

➤ P1 口（P1.0 ~ P1.7）：用做 I/O 口。

➤ P2 口（P2.0 ~ P2.7）：有两种功能。

☆ 通用 I/O 接口：无片外存储器时，P2 口可作通用 I/O 接口使用。

☆ 地址口：在访问外部存储器时，传送 ROM/RAM 高 8 位地址。

➤ P3 口（P3.0 ~ P3.7），其中 P3.0 是最低位双向 I/O 接口，有两种功能。

☆ 第一功能：通用 I/O 接口。

☆ 第二功能：用于串行口、中断源输入、计数器、片外 RAM 选通。

(2) 控制引脚（4 条）。

- $\overline{\text{ALE/PROG}}$ ：是地址锁存允许/编程脉冲输入信号线，配合 P0 口引脚的第二功能使用。
- $\overline{\text{PSEN}}$ ：是片外程序存储器 ROM 的开放信号引脚，用来输出访问片外程序存储器时的读选通信号
- $\overline{\text{EA/V}_{\text{PP}}}$ ：允许访问片外程序存储器/编程电源输入引脚，可以控制 8051 使用片内还是片外 ROM。
- $\overline{\text{RST/VPD}}$ ：复位/备用电源引脚。

(3) 时钟引脚（2 条）。MCS-51 单片机芯片内部有一个用于构成振荡器的高增益反相放大器，引脚 XTAL1 和 XTAL2 分别是此放大器的输入端和输出端。

- XTAL1：反相振荡放大器的输入引脚。
- XTAL2：反相振荡放大器的输出引脚。

(4) 电源引脚（2 条）。

- $V_{\text{CC}}$ ：单片机工作电源的输入引脚，接 +5V。
- $V_{\text{SS}}$ ：接地引脚。

2) 信号引脚的第二功能 信号引脚赋予双重功能，即“复用”。第二引脚功能主要集中在 P3 口。

(1) 常见的第二功能信号。

- P3 口线的第二功能：P3 口线都定义有第二功能，详见表 3-1。

表 3-1 P3 口的第二功能

引 脚	第 二 功 能
P3.0	RXD（串行输入口）
P3.1	TXD（串行输出口）
P3.2	$\overline{\text{INT0}}$ （外部中断 0 请求输入端）
P3.3	$\overline{\text{INT1}}$ （外部中断 1 请求输入端）
P3.4	T0（定时/计数器 0 请求脉冲输入端）
P3.5	T1（定时/计数器 1 请求脉冲输入端）
P3.6	$\overline{\text{WR}}$ （片外数据存储器写选通信号输出端）
P3.7	$\overline{\text{RD}}$ （片外数据存储器读选通信号输出端）

- EPROM 存储器程序固化所需要的信号：有内部 EPROM 的单片机芯片（如 87C51），为写入程序提供专门的编程脉冲和编程电源。它们也是由信号引脚以第二功能的形式提供的，即

☆ 编程脉冲：30 脚（ $\overline{\text{ALE/PROG}}$ ）。

☆ 编程电压（25V）：31 脚（ $\overline{\text{EA/VPP}}$ ）。

- 备用电源引入

MCS-51 单片机的备用电源也是以信号引脚第二功能的方式由 9 脚（ $\overline{\text{RST/VPD}}$ ）引入的。当电源发生故障，电压降低到下限值时，备用电源经由此端向内部 RAM 提供电压，以保护内部 RAM 中的信息不丢失。

(2) 引脚的第一、第二功能不会混淆。一个信号引脚，又是第一功能又是第二功能，会不会在使用时引起混乱和造成错误呢？不会的。这是因为：

- 对于各种型号的芯片，其引脚的第一功能信号是相同的，所不同的只在引脚的第二功能信号上。



- 对于 9、30 和 31 引脚，由于第一功能信号与第二功能信号是单片机在不同工作方式下的信号，因此不会发生使用上的矛盾。
- P3 口线的情况有所不同，它的第二功能信号都是单片机的重要控制信号。因此在实际使用时，总是先按需要优先选用它的第二功能，剩下不用的才作为通用 I/O 接口。

### 3. 51 单片机的时钟电路和时序

CPU 的操作需要精确的定时。MCS-51 单片机的时钟信号通常由两种电路形式得到：内部振荡方式和外部振荡方式。

将引脚 XTAL1 和 XTAL2 外接晶体振荡器（简称晶振）或陶瓷谐振器，就构成了内部振荡方式。由于单片机内部有一个高增益反相放大器，当外接晶振后，就构成了自激振荡器并产生振荡时钟脉冲。内部振荡方式的外部电路如图 3-3 所示，图中电容器  $C_1$ 、 $C_2$  起稳定振荡频率、快速起振的作用，其电容值一般为  $5 \sim 30\text{pF}$ 。晶振频率的典型值为  $12\text{MHz}$ ，采用  $6\text{MHz}$  的情况也比较多。内部振荡方式所得的时钟信号比较稳定，实用电路中使用较多。

外部振荡方式是把外部已有的时钟信号引入单片机内。这种方式适宜用来使单片机的时钟与外部信号保持同步。外部振荡方式的外部电路如图 3-4 所示，外部振荡信号由 XTAL2 引入，XTAL1 接地。为了提高输入电路的驱动能力，通常使外部信号经过一个带有上拉电阻的 TTL 反相门后接入 XTAL2。

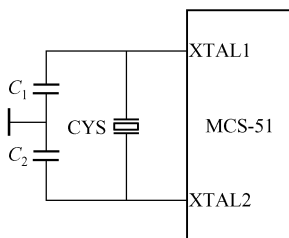


图 3-3 内部振荡方式

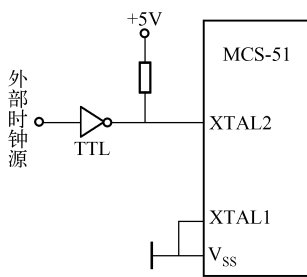


图 3-4 外部振荡方式

单片机以晶体振荡器的振荡周期（或外部引入的时钟周期）为最小的时序单位，片内的各种微操作都以此周期为时序基准。先了解一下相关的术语。

- 振荡周期：为单片机提供时钟信号的振荡源的周期。
  - 时钟周期：是振荡源信号经二分频后形成的时钟脉冲信号。
  - 机器周期：通常将完成一个基本操作所需的时间称为机器周期。
  - 指令周期：是指 CPU 执行一条指令所需要的时间。一个指令周期通常含有  $1 \sim 4$  个机器周期。
- 各种周期的相互关系如图 3-5 所示。

(1) 振荡周期：定时信号振荡器频率的倒数，用  $P$  表示。如  $6\text{MHz}$  时为  $1/6\mu\text{s}$ ， $12\text{MHz}$  时为  $1/12\mu\text{s}$ 。

(2) 时钟周期：对振荡周期二分频，它是振荡周期的两倍，又称状态周期，用  $S$  表示，其前半周期为  $P_1$ ，后半周期为  $P_2$ 。

(3) 机器周期：一个机器周期含 6 个时钟周期，分别用  $S_1 \sim S_6$  表示；含 12 个振荡周期，分别用  $S_1P_1$ 、 $S_1P_2$ 、 $\dots$ 、 $S_6P_2$  表示。

(4) 指令周期：完成一条指令所需要的时间。一个指令周期一般含 1 ~ 4 个机器周期。大部分指令是单字节单周期指令，少数是单字节双周期、双字节双周期指令，只有乘法和除法指令占用 4 个机器周期。

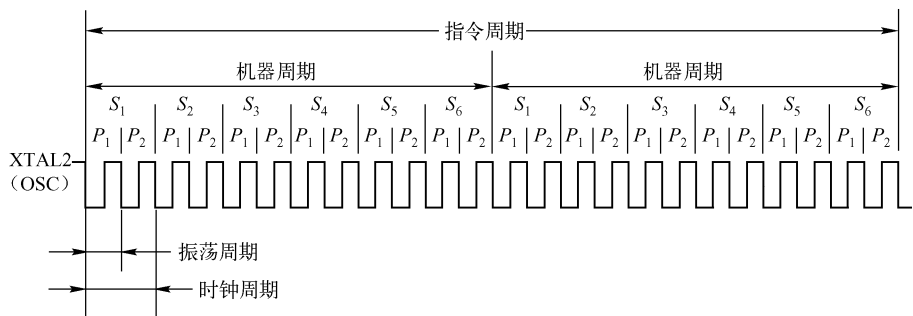


图 3-5 各种周期的相互关系

4. 51 单片机的复位状态及复位电路设计

当 MCS-51 系列单片机的复位引脚 RST（全称为 RESET）出现 2 个机器周期以上的高电平时，单片机就执行复位操作。如果 RST 持续为高电平，单片机就处于循环复位状态。根据应用的要求，复位操作通常有两种基本形式：上电复位和上电或开关复位。上电复位要求接通电源后，自动实现复位操作。常用的上电复位电路如图 3-6（a）所示，图中电容  $C_1$  和电阻  $R_1$  对电源 +5V 来说构成微分电路。上电后，保持 RST 一段高电平时间，由于单片机内等效电阻的作用，不用图中的电阻  $R_1$ ，也能达到上电复位的操作功能。

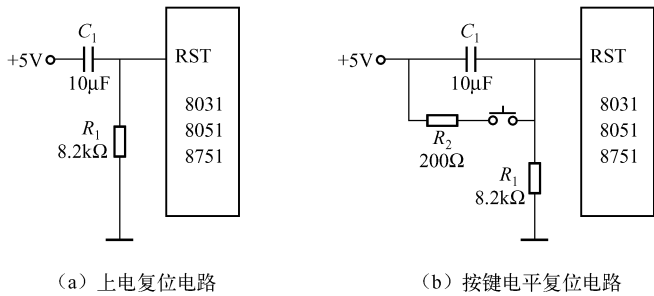


图 3-6 复位电路

上电或开关复位要求电源接通后，单片机自动复位，并且在单片机运行期间，用开关操作也能使单片机复位。常用的开关复位电路如图 3-6（b）所示。上电后，由于电容  $C_1$  的充电和反相门的作用，使 RST 持续一段时间的高电平。当单片机已在运行中时，按下复位键 K 后松开，也能使 RST 保持一段时间的高电平，从而实现上电或开关复位的操作。

单片机复位是使 CPU 和系统中的其他功能部件都处在一个确定的初始状态，并从这个状态开始工作，例如，复位后  $PC = 0000H$ ，使单片机从第一个单元取指令。无论是在单片机刚开始接上电源时，还是断电后或发生故障后都要复位。51 单片机复位后机器的初始状态即各寄存器的状态为：PC 之外，复位操作还对其他一些特殊功能寄存器有影响，它们的复位状态如表 3-2 所示。

表 3-2 复位后特殊功能寄存器的状态

寄 存 器	复位时的内容	寄 存 器	复位时的内容
PC	0000H	TCON	0 × 000000B
ACC	00H	TL0	00H
B	00H	TH0	00H
PSW	00H	TH1	00H
SP	07H	TH1	00H
DPTR	0000H	SCON	00H
P0 ~ P3	FFH	SBUF	不确定
TMOD	× × 000000B	PCON	0 × × × 0000B

## 5. 存储器、特殊功能寄存器及位地址空间

1) 51 单片机存储器 计算机的存储器有两种结构：一种称为哈佛结构，即程序存储器和数据存储器分开，相互独立；另一种称为普林斯顿结构，即程序存储器和数据存储器是统一的，地址空间统一编址。MCS-51 单片机属于哈佛结构，程序存储器和数据存储器分开。8051 的存储器在物理结构上分为程序存储空间和数据存储空间，共有 4 个存储空间：片内程序存储器和片内数据存储器及片外程序存储器和片外数据存储器，如图 3-7 所示。

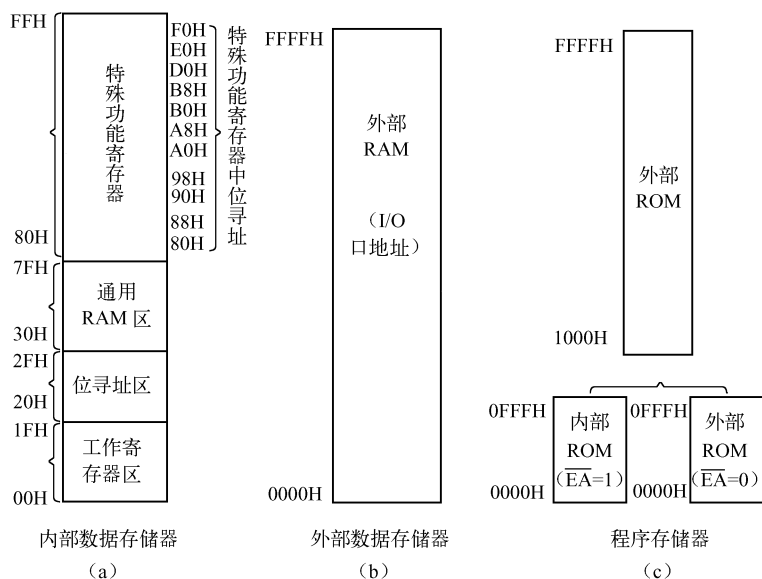


图 3-7 51 单片机存储器空间

(1) MCS-51 内部程序存储器。MCS-51 的程序存储器用于存放编好的程序和表格常数。8051 片内有 4KB 的 ROM，8751 片内有 4KB 的 EPROM，8031 片内无程序存储器。MCS-51 的片外最多能扩展 64KB 程序存储器，片内外的 ROM 是统一编址的。如 EA 端保持高电平，8051 的程序计数器 PC 在 0000H ~ 0FFFH 地址范围内（即前 4KB 地址）执行片内 ROM 中的程序，当 PC 在 1000H ~ FFFFH 地址范围时，自动执行片外程序存储器中的程序；当 EA 端保持低电平时，只能寻址外部程序存储器，片外存储器可以从 0000H 开始编址。

MCS-51 的程序存储器中有些单元具有特殊功能，使用时应予以注意。

其中一组特殊单元是 0000H ~ 0002H。系统复位后，(PC) = 0000H，单片机从 0000H 单元开始取指令执行程序。如果程序不从 0000H 单元开始，应在这三个单元中存放一条无条件转移指令，以便直接转去执行指定的程序。

还有一组特殊单元是 0003H ~ 002AH，共 40 个单元。这 40 个单元被均匀地分为 5 段，作为 5 个中断源的中断地址区，如表 3-3 所示。

表 3-3 51 单片机中断源的中断地址区

0003H~000AH	外部中断 0 中断地址区
000BH~0012H	定时/计数器 0 中断地址区
0013H~001AH	外部中断 1 中断地址区
001BH~0022H	定时/计数器 1 中断地址区
0023H~002AH	串行中断地址区

中断响应后，按中断种类，自动转到各中断区的首地址去执行程序，因此在中断地址区中理应存放中断服务程序。但通常情况下，8 个单元难以存下一个完整的中断服务程序，因此通常从中断地址区首地址开始存放一条无条件转移指令，以便中断响应后，通过中断地址区再转到中断服务程序的实际入口地址。

(2) MCS-51 内部数据存储器。8051 的内部 RAM 共有 256 个单元，通常把这 256 个单元按其功能划分为两部分：低 128 单元（单元地址 00H ~ 7FH）和高 128 单元（单元地址 80H ~ FFH）。如图 3-7（a）所示为低 128 单元的配置图。

① 寄存器区：8051 共有 4 组寄存器，每组 8 个寄存单元（各为 8），各组都以 R0 ~ R7 作为寄存单元编号。寄存器常用于存放操作数中间结果等，由于它们的功能及使用不作预先规定，因此称为通用寄存器，有时也叫工作寄存器。4 组通用寄存器占据内部 RAM 的 00H ~ 1FH 单元地址。

在任一时刻，CPU 只能使用其中的一组寄存器，并且把正在使用的那组寄存器称为当前寄存器组。到底是哪一组，由程序状态字寄存器 PSW 中 RS1、RS0 位的状态组合来决定，如表 3-4 所示。

表 3-4 51 单片机的通用寄存器

RS0	RS1	寄存器组	RAM
0	0	第 0 组	00H~07H
0	1	第 1 组	08H~0FH
1	0	第 2 组	10H~17H
1	1	第 3 组	18H~1FH

② 位寻址区：内部 RAM 的 20H ~ 2FH 单元，既可作为一般 RAM 单元使用，进行字节操作，也可对单元中的每一位进行位操作，因此把该区称为位寻址区。位寻址区共有 16 个 RAM 单元，计 128 位，地址为 00H ~ 7FH，如表 3-5 所示。MCS-51 具有布尔处理机功能，这个位寻址区可以构成布尔处理机的存储空间。这种位寻址能力是 MCS-51 的一个重要特点。

表 3-5 RAM 位寻址区地址表

单元地址	MSB 位地址 LSB							
2FH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H
2EH	77H	76H	75H	74H	73H	72H	71H	70H
2DH	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H
2CH	67H	66H	65H	64H	63H	62H	61H	60H
2BH	5FH	5EH	5DH	5CH	5BH	5AH	59H	58H
2AH	57H	56H	55H	54H	53H	52H	51H	50H
29H	4FH	4EH	4DH	4CH	4BH	4AH	49H	48H
28H	47H	46H	45H	44H	43H	42H	41H	40H
27H	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H
26H	37H	36H	35H	34H	33H	32H	31H	30H
25H	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H
24H	27H	26H	25H	24H	23H	22H	21H	20H
23H	1FH	1EH	1DH	1CH	1BH	1AH	19H	18H
22H	17H	16H	15H	14H	13H	12H	11H	10H
21H	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H
20H	07H	06H	05H	04H	03H	02H	01H	00H

③ 用户 RAM 区：在内部 RAM 低 128 单元中，通用寄存器占去 32 个单元，位寻址区占去 16 个单元，剩下 80 个单元，这就是供用户使用的一般 RAM 区，其单元地址为 30H ~ 7FH。对用户 RAM 区的使用没有任何规定或限制，但在一般应用中常把堆栈开辟在此区中。

④ 内部数据存储器高 128 单元：内部 RAM 的高 128 单元是供给专用寄存器使用的，其单元地址为 80H ~ FFH。因这些寄存器的功能已作专门规定，故称为专用寄存器（Special Function Register），也可称为特殊功能寄存器。MCS-51 有 22 个专用寄存器，其中 21 个为可寻址，如表 3-6 所示。

表 3-6 MCS-51 专用寄存器

符 号	名 称	地 址
* ACC	累加器	0E0H
* B	B 寄存器	0F0H
* PSW	程序状态字	0D0H
DPL	数据指针低 8 位	82H
DPH	数据指针高 8 位	83H
SP	栈指针	81H
* P0	P0 口锁存寄存器	80H
* P1	P1 口锁存寄存器	90H
* P2	P2 口锁存寄存器	0A0H
* P3	P3 口锁存寄存器	0B0H
* IP	中断优先级控制寄存器	0B8H

续表

符 号	名 称	地 址
* IE	中断允许控制寄存器	0A8H
TMOD	定时/计数器工作方式寄存器	89H
* TCON	定时控制寄存器	88H
TH0	定时器 0（高字节）	8CH
TL0	定时器 0（低字节）	8AH
TH1	定时器 1（高字节）	8DH
TL1	定时器 1（低字节）	8BH
* SCON	串行口控制寄存器	98H
SBUF	串行数据缓冲器	99H
PCON	电源控制及波特率选择寄存器	97H

注：标有 \* 标记的 11 个特殊功能寄存器，其字节地址正好能被 8 除整，它们的字节单元的每一位可以按位寻址和操作。

下面将介绍 22 个专用寄存器中的 4 个，其余的专用寄存器在以后的章节中陆续介绍。

- 程序计数器（PC—Program Counter）：PC 是一个 16 位的计数器，其内容为将要执行的指令地址，寻址范围达 64KB。PC 有自动加 1 功能，从而实现程序的顺序执行。PC 没有地址，是不可寻址的，因此用户无法对它进行读/写。但可以通过转移、调用、返回等指令改变其内容，以实现程序的转移。
- 累加器（ACC—Accumulator）：累加器为 8 位寄存器，是最常用的专用寄存器，功能较多，地位重要。它既可用于存放操作数，也可用来存放运算的中间结果。MCS-51 单片机中大部分单操作数指令的操作数就取自累加器，许多双操作数指令中的一个操作数也取自累加器。
- B 寄存器：B 寄存器是一个 8 位寄存器，主要用于乘除运算。乘法运算时，B 是乘数；乘法操作后，乘积的高 8 位存于 B 中。除法运算时，B 是除数；除法操作后，余数存于 B 中。此外，B 寄存器也可作为一般数据寄存器使用。
- 程序状态字（Program Status Word, PSW）：程序状态字是一个 8 位寄存器，用于寄存程序运行的状态信息。其中有些位状态是根据程序执行结果由硬件自动设置的，而有些位状态则使用软件方法设定。PSW 的位状态可以用专门指令进行测试，也可以用指令读出。一些条件转移指令将根据 PSW 有关位的状态进行程序转移。PSW 各位的定义如表 3-7 所示。

表 3-7 PSW 寄存器的位状态

位 序	PSW. 7	PSW. 6	PSW. 5	PSW. 4	PSW. 3	PSW. 2	PSW. 1	PSW. 0
位标志	CY	AC	F0	RS1	RS0	OV	/	P

- ☆ 进位标志 CY 或 C（PSW. 7）：其功能有二，一是存放算术运算的进位标志；二是在位操作中作累加位使用。在位的传送、位与、位或等位操作中，都要使用进位标志。在执行某些算术操作类指令时，可被硬件或软件置位或清零。例如，在 8 位加法运算时，若运算结果最高位 D7 有进位，则 C=1，否则 C=0；又如，在 8 位减法运算时，若运算结果的最高位 D7 有借位，则 C=1，否则 C=0。半数以上的位操作指令与 C 有关，可见位处理时，它起着“位累加器”的作用。
- ☆ 辅助进位标志 AC（PSW. 6）：8 位加法运算时，如果低半字节的最高位 D3 有进位，则 AC=1，否则 AC=0；在 8 位减法运算时，如果低半字节的最高位 D3 有借位，则 AC=1，否则 AC=0。AC 在作为 BCD 码运算而进行二至十进制调整时有用。
- ☆ 软件标志位 F0（PSW. 5）：这是用户定义的一个状态标志，可通过软件对它置位、清零；在编程时，也常测试其是否建起而进行程序分支。

- ☆ 工作寄存器组选择位 RS1、RS0 (PSW.4、PSW.3): 可借软件置位或清零, 以选定 4 个工作寄存器中的一个组投入工作。
- ☆ 溢出标志位 OV (PSW.2): 作为符号数在进行加法、减法时由硬件置位或清零, 以指示运算结果是否溢出, 运算结果放累加器。OV=1, 反映它已超出累加器, 以补码形式表示一个有符号数的范围 (-128 ~ +127)。在乘法运算中, OV=1, 表示乘积超过 255, 即乘积分别在 B 与 A 中; 反之, OV=0, 表示乘积只在 A 中。在除法运算中, OV=1, 表示除数为零, 除法不能进行; 反之, OV=0, 表示除数不为零, 除法可正常进行。
- ☆ 奇偶标志 P (PSW.0): 每执行一条指令, 单片机都能根据 A 中 1 的个数的奇偶自动令 P 置位或清零, 奇数为 1, 偶数为 0。此标志对串行通信的数据传输非常有用, 通过奇偶校验可检验传输的可靠性。
- ☆ 数据指针 (DPTR)
 

数据指针为 16 位寄存器, 它是 MCS-51 中唯一一个 16 位寄存器。编程时, DPTR 既可以按 16 位寄存器使用, 也可以按两个 8 位寄存器分开使用。即

DPH——DPTR 高位字节;

DPL——DPTR 低位字节。

DPTR 通常在访问外部数据存储时作为地址指针使用, 由于外部数据存储器的寻址范围为 64KB, 故把 DPTR 设计为 16 位。

对专用寄存器的字节寻址问题作如下几点说明。

- 21 个可字节寻址的专用寄存器是不连续地分散在内部 RAM 高 128 单元之中, 尽管还余有许多空闲地址, 但用户并不能使用。
- 程序计数器 PC 不占据 RAM 单元, 它在物理上是独立的, 因此是不可寻址的寄存器。
- 对专用寄存器只能使用直接寻址方式, 书写时既可使用寄存器符号, 也可使用寄存器。

## 6. 输入/输出 (I/O) 口

单片机芯片内还有一项主要内容就是并行 I/O 口。MCS-51 共有 4 个 8 位的并行 I/O 口, 分别记做 P0、P1、P2、P3。每个口都包含一个锁存器、一个输出驱动器和输入缓冲器。实际上, 它们已被归入专用寄存器之列, 并且具有字节寻址和位寻址功能。

在访问片外扩展存储器时, 低 8 位地址和数据由 P0 口分时传送, 高 8 位地址由 P2 口传送。在无片外扩展存储器的系统中, 这 4 个口的每一位均可作为双向的 I/O 端口使用。I/O 口内部结构如图 3-8 所示。

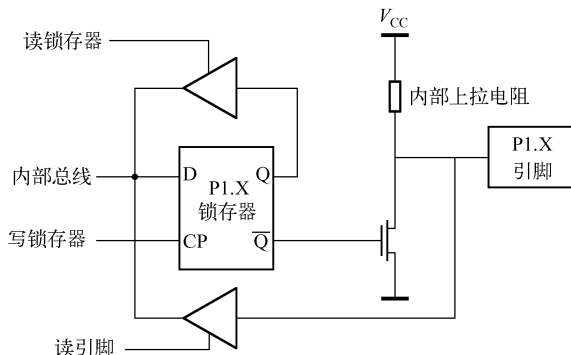


图 3-8 I/O 口内部结构图



### 3.2 51 单片机的中断系统

在 CPU 与外设交换信息时，存在着一个快速的 CPU 与慢速的外设之间的矛盾；单片机在某一时刻只能处理一个任务，当多个任务同时要求单片机处理时，这一要求应该如何实现呢？为解决这些问题，发展了中断的概念。

中断现象在现实生活中也会经常遇到，例如，你在看书——手机响了——你在书上作个记号——你接电话和对方聊天——谈话结束——从书上的记号处继续看书。这就是一个中断过程。通过中断，你一个人在一个特定的时刻，同时完成了看书和打电话两件事情。用计算机语言来描述，所谓的中断就是，当 CPU 正在处理某项事务时，如果外界或内部发生了紧急事件，要求 CPU 暂停正在处理的工作而去处理这个紧急事件，待处理完后，再回到原来中断的地方，继续执行原来被中断的程序，这个过程称做中断。

从中断的定义可以看到中断应具备中断源、中断响应、中断返回这样三个要素。中断源发出中断请求，单片机对中断请求进行响应，当中断响应完成后应进行中断返回，返回中断的地方继续执行原来被中断的程序。

#### 1. 中断请求源

对于中断系统来说，引起中断的事件称为中断源；由中断源向 CPU 发出的请求中断的信号称为中断请求信号；CPU 中止现行程序执行的位置称为中断断点；中断断点处的程序位置称为中断现场；由中断服务程序返回原来程序的过程称为中断返回；CPU 接收中断请求而中止现行程序，转去为中断源服务称为中断响应。中断源越多，表示 CPU 处理随机事务的能力越强。MCS-51 单片机具有 5 个中断源（52 系列有 6 个中断源），两个优先级，可以实现中断处理程序的两级嵌套，中断处理能力较强。这 5 个中断源是：

- 两个外部中断（INT0 和 INT1，由 P3.2 和 P3.3 输入，中断标志分别为 IE0 和 IE1）；
- 两个定时/计数器溢出中断（T0 和 T1，外部计数/内部定时，中断标志分别为 TF0 和 TF1）；
- 串行口中断（中断标志分别为 TI 和 RI）。

MCS-51 单片机中断系统的内部结构示意图如图 3-9 所示。

#### 2. 中断控制

中断控制的内容包括 CPU 中断的开放与禁止、中断允许、中断优先级等控制。中断控制是通过两个特殊功能寄存器 IE（中断允许寄存器）和 IP（中断优先级控制寄存器）写入控制字（编程）实现的。此外，中断的控制还与特殊功能寄存器 TCON（定时/计数器控制寄存器）、SCON（串行口控制寄存器）有关。

1) 中断允许寄存器 IE IE 为 8 位的 SFR，片内 RAM 地址为 0A8H（有 6 位可位寻址），其各位的功用见表 3-8。

表 3-8 中断允许寄存器的位功用

D7	D6	D5	D4	D3	D2	D1	D0
EA	x	x	ES	ET1	EX1	ET0	EX0



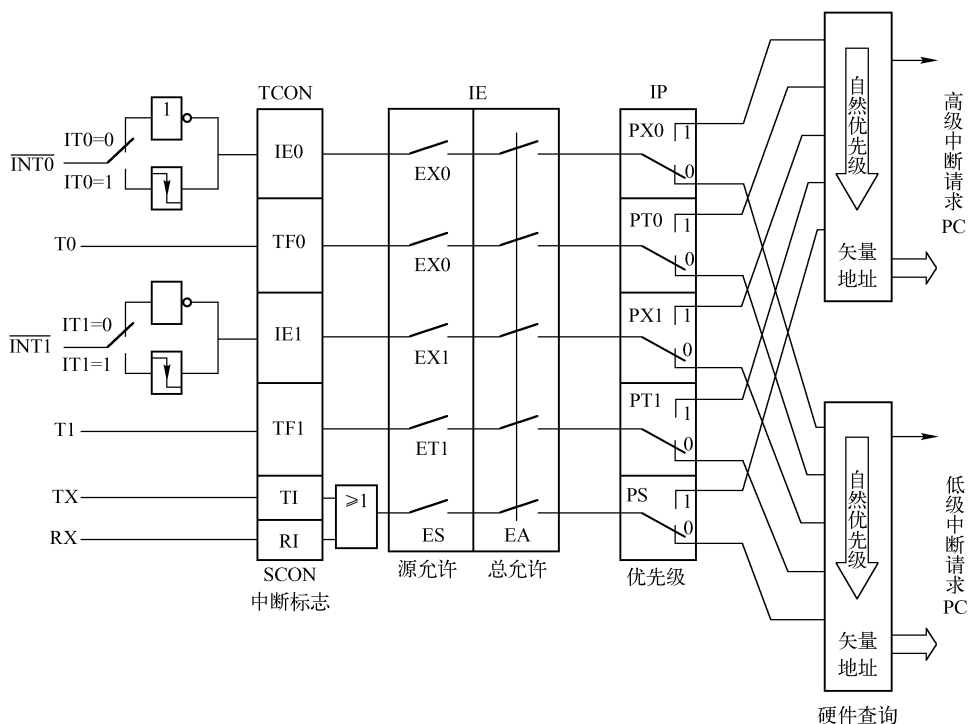


图 3-9 MCS-51 单片机中断系统的内部结构示意图

- ES：允许或禁止串行口中断。若 ES=0，则禁止 CPU 响应串行口中断；否则允许串行口中断。
- ET1：允许或禁止定时/计数器 1 溢出中断。若 ET1=0，则禁止定时/计数器 1 中断；否则允许定时/计数器 1 中断。
- EX1：允许或禁止外部中断 1。若 EX1=0，则禁止外部中断 1；否则允许外部中断 1。
- 如 ET0：允许或禁止定时/计数器 0 溢出中断。若 ET0=0，则禁止定时/计数器 0 中断；否则允许定时/计数器 0 中断。
- 如 EX0：允许或禁止外部中断 0。若 EX0=0，则禁止外部中断 0；否则允许外部中断 0。
- 如 EA：开放或禁止所有中断，为 CPU 中断控制的“总开关”。若 EA=0，则不响应任何中断；EA=1，则每种中断的响应分别由上述各自允许位的置位或清除来确定允许或禁止。

**2) 中断优先级控制寄存器 IP** 当单片机同时接收到两种或两种以上的中断请求时，响应哪一种中断除与 IE 的设置有关外，主要受中断优先级控制寄存器 IP（片内 RAM 地址为 0B8H，所有 8 位可位寻址）的控制。MCS-51 的中断优先级有两种：高优先级和低优先级。

任何一种中断源的优先级都可通过对 IP 的编程（使相应位置位或清零）而使之设置成高、低优先级中的一种。IP 各位的功用如表 3-9 所示。

表 3-9 中断优先级控制寄存器 IP 的位功用

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	PS	PT1	PX1	PT0	PX0

- PS：串行口中断优先级控制位。PS=1 为高优先级，PS=0 为低优先级。
- PT1：定时/计数器 1 中断优先级控制位。PT1=1 为高优先级，PT1=0 为低优先级。
- PX1：外部中断 1 中断优先级控制位。PX1=1 为高优先级，PX1=0 为低优先级。

- PT0：定时/计数器 0 中断优先级控制位。PT0=1 为高优先级，PT0=0 为低优先级。
- PX0：外部中断 0 中断优先级控制位。PX0=1 为高优先级，PX0=0 为低优先级。

【说明】单片机复位之后，IE 和 IP 均被清零。

MCS-51 单片机对中断优先级的处理原则如下。

- 不同级的中断源同时申请中断时：先高后低。
- 处理低级中断又收到高级中断请求时：停低转高。
- 处理高级中断却收到低级中断请求时：高不睬低。
- 同一级的中断源同时申请中断时：事先预定。
- 对于同一优先级，单片机对其中断的次序安排如下：

高—————→低  
INT0→TF0→INT1→TF1→RI 或 TI

3) 定时/计数器控制寄存器 (TCON) TCON 为定时/计数器控制寄存器，其字节映像地址为 88H，可位寻址，各位的地址为 88H ~ 8FH。它除了控制定时/计数器 T0、T1 的溢出中断外，还控制着两个外部中断源的触发方式和锁存两个外部中断源的中断请求标志。该寄存器的位功用如表 3-10 所示。

表 3-10 定时/计数器控制寄存器 (TCON) 的位功用

位 地 址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
位符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- IT0 和 IT1——外部中断请求触发方式控制位。
  - ☆ IT0 (IT1) = 1，脉冲触发方式，下降沿有效；
  - ☆ IT0 (IT1) = 0，电平触发方式，低电平有效。
- IE0 和 IE1——外中断请求标志位。当 CPU 采样到  $\overline{\text{INT0}}$  (或  $\overline{\text{INT1}}$ ) 端出现有效中断请求时，IE0 (IE1) 位由硬件置 1。当中断响应完成转向中断服务程序时，由硬件将 IE0 (或 IE1) 清 0。
- TR0 和 TR1——定时器运行控制位。
  - ☆ TR0 (TR1) = 0，定时/计数器不工作；
  - ☆ TR0 (TR1) = 1，定时/计数器开始工作。
- TF0 和 TF1——计数溢出标志位。当计数器产生计数溢出时，相应的溢出标志位由硬件置 1。当转向中断服务时，再由硬件自动清 0。计数溢出标志位的使用有两种情况：采用中断方式时，作为中断请求标志位使用；采用查询方式时，作为查询状态位使用。

4) 串行口控制寄存器 (SCON) SCON 为串行口控制寄存器，其字节映像地址为 98H，也可以进行位寻址，各位的地址为 98H ~ 9FH。串行口的接收和发送数据中断请求标志位 (RI、TI) 被锁存在串行口控制寄存器 SCON 中，寄存器的位功用如表 3-11 所示。

表 3-11 串行口控制寄存器 SCON 的位功用

位 地 址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
位符号	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SCON 中与中断有关的控制位共有两位。

- TI——串行口发送中断请求标志位。当发送完一帧串行数据后，由硬件置 1；在转向中断服务程序后，用软件清 0。

- RI——串行口接收中断请求标志位。当接收完一帧串行数据后，由硬件置 1；在转向中断服务程序后，用软件清 0。串行中断请求由 TI 和 RI 的逻辑或得到。就是说，无论是发送标志还是接收标志，都会产生串行中断请求。

### 3. 中断的响应过程

中断响应过程为：中断源发出中断请求→对中断请求进行响应→执行中断服务程序→返回主程序。这个过程可分为三个阶段来完成。

1) 中断采样 对于外部中断请求，中断请求信号来自于单片机外部，计算机要想知道有没有中断请求发生，必须对信号进行采样。

- 电平触发方式的外中断请求（IT0/IT1 = 0）采样到高电平时，表明没有中断请求，IE0 或 IE1 继续为 0；采样到低电平时，IE0/IE1 由硬件自动置 1，表明有外中断请求发生。
- 脉冲触发方式的外中断请求（IT0/IT1 = 1）在相邻的机器周期采样到的电平由高电平变为低电平时，IE0/IE1 由硬件自动置 1，否则为 0。

2) 中断查询 由 CPU 测试 TCON 和 SCON 中各个中断标志位的状态，确定有哪个中断源发生请求，查询时按优先级顺序进行，即先查询高优先级再查询低优先级。如果同级，按以下顺序查询：

$$\overline{\text{INT0}} \rightarrow \text{T1} \rightarrow \overline{\text{INT1}} \rightarrow \text{T1} \rightarrow \text{S}$$

如果查询到有标志位为 1，表明有中断请求发生，接着就从相邻的下一机器周期开始进行中断响应。

3) 中断响应 当 CPU 查询到中断请求时，由硬件自动产生一条 LCALL 指令，LCALL 指令执行时，首先将 PC 内容压入堆栈进行断点保护，再把中断入口地址装入 PC，使程序转向相应的中断区入口地址。LCALL 指令的形式如下：

LCALL addr16;addr16:中断入口地址

入口地址已由系统设定，如表 3-12 所示。

表 3-12 中断源的入口地址

中 断 源	入 口 地 址
$\overline{\text{INT0}}$	0003H
T0	000BH
$\overline{\text{INT1}}$	0013H
T1	001BH
S	0023H


从表中可以看出，每个中断区只有 8 个单元，很难安排下一个中断程序，一般是在中断入口地址处加一条跳转指令，跳转到用户的服务程序入口。编写中断服务程序的格式一般如下：

```
ORG    0000H
SJMP   MAIN
ORG    0003H
AJMP   INTSER
MAIN:  ...
```

```
HERE: SJMP  HERE
INTSER                ;中断响应程序
RETI
```

并不是所有的请求都会被响应，当遇到下列情况之一时不响应这些中断请求：

- (1) CPU 正在处理一个同级或高级的中断服务；
- (2) 当前指令还没有执行完毕；
- (3) 当前指令是 RET、RETI 或是访问 IP、IE 的指令，执行完这些指令后，还必须再执行一条指令，才响应中断请求。

 注意：MCS-51 单片机对中断查询结果不作记忆，当有新的查询结果出现时，因为以上原因而被拖延的查询结果将不复存在，其中断请求也就不能再被响应了。

**4) 中断请求的撤销** 中断响应后，TCON 和 SCON 的中断请求标志位应及时撤销。否则意味着中断请求仍然存在，有可能造成中断的重复查询和响应，因此需要在中断响应完成后，撤销其中断标志。

- (1) 定时中断请求的撤销。硬件自动将 TF0 (TF1) 清零，不需要用户参与。
- (2) 串行中断请求的撤销，需要软件清零。
- (3) 外部中断请求的撤销。

➤ 脉冲触发方式的外中断请求撤销：中断标志位的清零是自动的，脉冲信号过后就不存在了，因此其撤销是自动的。

➤ 电平触发方式的外中断请求撤销：中断标志位的清零是自动的，但是如果低电平持续存在，在以后的机器周期采样时，又会将中断请求标志位 (IE0/IE1) 置位。为此，需要外加电路，将中断请求信号从低电平强制为高电平。

**5) 中断响应时间** MCS-51 单片机的中断响应延迟时间，取决于其他中断服务程序是否在进行，或取决于正在执行的是什么样的指令。单中断系统的中断响应时间为 3 ~ 8 个机器周期。无论是哪一种原因引起的误差，在精确定时的应用场合必须考虑它们的影响，以确保精确的定时控制。根据定时中断的不同应用情况，应选择不同的精确定时编程方法。



## 3.3 51 单片机的串行接口及串行通信技术

### 1. 串行通信基础

在计算机系统中，CPU 和外部通信有两种通信方式：并行通信和串行通信。并行通信，即数据的各位同时传送；串行通信，即数据一位一位顺序传送。图 3-10 为这两种通信方式的示意图。

上述两种基本通信方式比较起来，串行通信能够节省传输线，特别是在数据位数很多和远距离传送数据时，这一优点更为突出；串行通信方式的主要缺点是传送速度比并行通信慢。

**1) 串行通信的分类** 按照串行数据的时钟控制方式，串行通信可分为同步通信和异步通信两类。

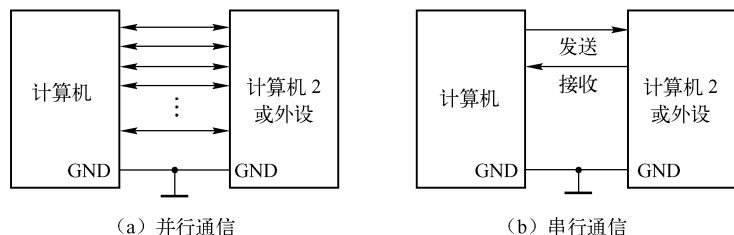


图 3-10 两种通信方式的示意图

(1) 异步通信 (Asynchronous Communication)。在异步通信中, 数据通常是以字符为单位组成字符帧传送的。字符帧由发送端一帧一帧地发送, 每一帧数据都是低位在前, 高位在后, 通过传输线被接收端一帧一帧地接收。发送端和接收端可以由各自独立的时钟来控制数据的发送和接收, 这两个时钟彼此独立, 互不同步。

在异步通信中, 接收端是依靠字符帧格式来判断发送端是何时开始发送、何时结束发送的。字符帧格式是异步通信的一个重要指标。

① 字符帧 (Character Frame): 也叫数据帧, 由起始位、数据位、奇偶校验位和停止位 4 部分组成, 如图 3-11 所示。

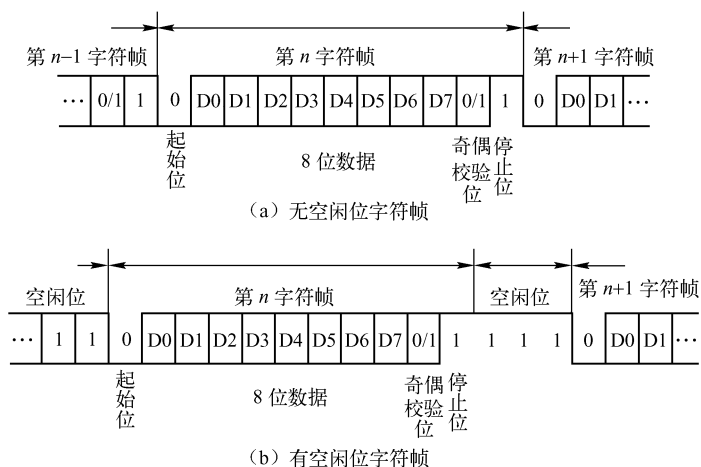


图 3-11 异步通信的字符帧格式

- 起始位: 位于字符帧开头, 只占一位, 为逻辑 0 低电平, 用于向接收设备表示发送端开始发送一帧信息。
- 数据位: 紧跟起始位之后, 用户根据情况可取 5 位、6 位、7 位或 8 位, 低位在前、高位在后。
- 奇偶校验位: 位于数据位之后, 仅占一位, 用来表征串行通信中采用奇校验还是偶校验, 由用户决定。
- 停止位: 位于字符帧最后, 为逻辑 1 高电平。通常可取 1 位、1.5 位或 2 位, 用于向接收端表示一帧字符信息已经发送完, 也为发送下一帧做准备。

在串行通信中, 两相邻字符帧之间可以没有空闲位, 也可以有若干空闲位, 这由用户来决定。图 3-11 (b) 表示有 3 个空闲位的字符帧格式。

② 波特率 (baud rate): 异步通信的另一个重要指标为波特率。

波特率为每秒钟传送二进制数码的位数, 也叫比特数, 单位为 b/s, 即位/秒。波特率

用于表征数据传输的速度，波特率越高，数据传输速度越快。但波特率和字符的实际传输速率不同，字符的实际传输速率是每秒内所传字符帧的帧数，和字符帧格式有关。

通常，异步通信的波特率为 50 ~ 9 600b/s。

异步通信的优点是不需要传送同步时钟，字符帧长度不受限制，故设备简单；缺点是字符帧中因包含起始位和停止位而降低了有效数据的传输速率。

(2) 同步通信 (Synchronous Communication)。同步通信是一种连续串行传送数据的通信方式，一次通信只传输一帧信息。这里的信息帧和异步通信的字符帧不同，通常有若干个数据字符，如图 3-12 所示。图 3-12 (a) 为单同步字符帧结构，图 3-12 (b) 为双同步字符帧结构，它们均由同步字符、数据字符和校验字符 CRC 三部分组成。在同步通信中，同步字符可以采用统一的标准格式，也可以由用户约定。

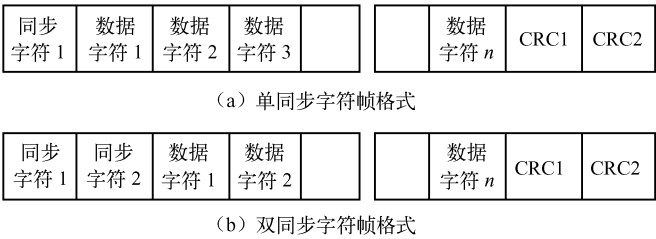


图 3-12 同步通信的字符帧格式

同步通信的优点是数据传输速率较高，通常可达 56 000b/s 或更高，缺点是要求发送时钟和接收时钟必须保持严格同步。

2) 串行通信的制式 在串行通信中数据是在两个站之间进行传送的，按照数据传送方向，串行通信可分为单工 (simplex)、半双工 (half duplex) 和全双工 (full duplex) 三种制式。图 3-13 为三种制式的示意图。

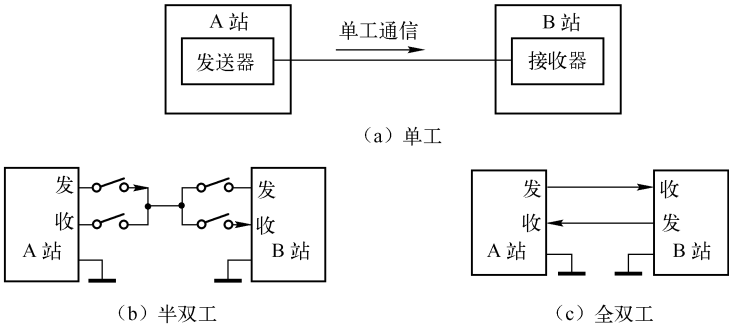


图 3-13 单工、半双工和全双工三种制式示意图

在单工制式下，通信线的一端接发送器，一端接接收器，数据只能按照一个固定的方向传送，如图 3-13 (a) 所示。

在半双工制式下，系统的每个通信设备都由一个发送器和一个接收器组成，如图 3-13 (b) 所示。在这种制式下，数据能从 A 站传送到 B 站，也能从 B 站传送到 A 站，但是不能同时两个方向上传送，即只能一端发送，一端接收。其收发开关一般是由软件控制的电子开关。

全双工通信系统的每一端都有发送器和接收器，可以同时发送和接收，即数据可以在两个方向上同时传送，如图 3-13 (c) 所示。

在实际应用中，尽管多数串行通信接口电路都具有全双工功能，但一般情况只工作于半双工制式下，因为这种用法简单、实用。

**3) 串行通信的接口电路** 串行接口电路的种类和型号很多。能够完成异步通信的硬件电路称为 UART，即通用异步接收/发送器 (Universal Asynchronous Receiver/Transmitter)；能够完成同步通信的硬件电路称为 USRT (Universal Synchronous Receiver/Transmitter)；既能完成异步又能完成同步通信的硬件电路称为 USART (Universal Synchronous Asynchronous Receiver/Transmitter)。

从本质上说，所有的串行接口电路都是以并行数据形式与 CPU 接口，以串行数据形式与外部逻辑接口。它们的基本功能都是从外部逻辑接收串行数据，转换成并行数据后传送给 CPU，或从 CPU 接收并行数据，转换成串行数据后输出到外部逻辑。

## 2. MCS-51 单片机串行接口

MCS-51 内部有一个可编程全双工串行通信接口，它具有 UART 的全部功能，该接口不仅可以同时进行数据的接收和发送，也可作同步移位寄存器使用。该串行口有 4 种工作方式，帧格式有 8 位、10 位和 11 位，并能设置各种波特率。

串行口的结构如图 3-14 所示。

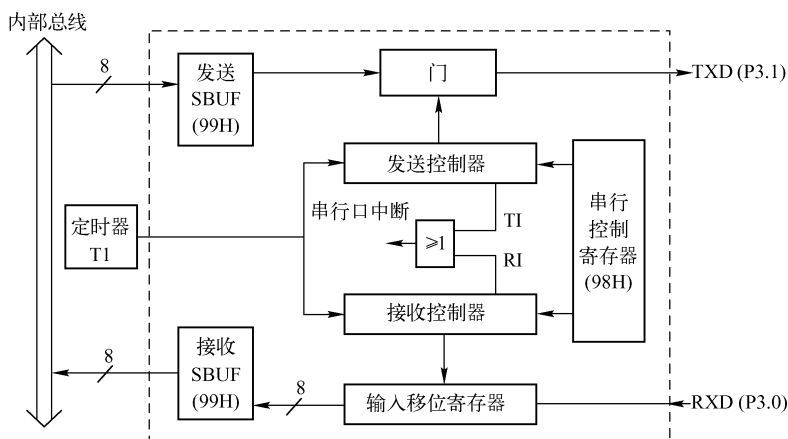


图 3-14 串行口的结构示意图

- 两个物理上独立、同名的接收/发送缓冲寄存器 SBUF：指令 MOV SBUF, A 启动一次数据发送；指令 MOV A, SBUF 完成一次数据接收。
- 输入/输出移位寄存器和控制器等。
- 两个 SFR 寄存器 SCON 和 PCON，用于串行口的初始化编程。
- 接收/发送数据，无论是否采用中断方式工作，每接收/发送一个数据都必须用指令对 RI/TI 清零，以备下一次收/发。

与 MCS-51 串行口有关的特殊功能寄存器有 SBUF、SCON、PCON，下面分别对它们进行详细的讨论。

**1) 串行口数据缓冲器 SBUF** SBUF 是两个在物理上独立的接收、发送寄存器，一个用于存放接收到的数据，另一个用于存放欲发送的数据，可同时发送和接收数据。两个缓冲器

共用一个地址 99H，通过对 SBUF 的读、写指令来区别是对接收缓冲器还是发送缓冲器进行操作。CPU 在写 SBUF 时，就是修改发送缓冲器；读 SBUF，就是读取接收缓冲器的内容。接收或发送数据，是通过串行口对外的两条独立收发信号线 RXD (P3.0)、TXD (P3.1) 实现的，因此可以同时发送、接收数据，为全双工制式。

2) 串行口控制寄存器 SCON 收发双方都有对 SCON 的编程，SCON 用来控制串行口的工作方式和状态，可以位寻址，字节地址为 98H。单片机复位时，所有位全为 0。其格式如表 3-13 所示。

表 3-13 SCON 寄存器各位定义

9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- SM0、SM1：串行方式选择位。将在串行口的工作方式中进行讲解。
- SM2：多机通信控制位，用于方式 2 和方式 3 中。在方式 2 和方式 3 处于接收时，若 SM2 = 1 且接收到的第 9 位数据 RB8 为 0，不激活 RI；若 SM2 = 1 且 RB8 = 1，则置 RI = 1。在方式 2 和方式 3 处于接收或发送方式时，若 SM2 = 0，不论接收到的第 9 位 RB8 为 0 还是为 1，TI、RI 都以正常方式被激活。在方式 1 处于接收时，若 SM2 = 1，则只有接收到有效的停止位后，RI 置 1。在方式 0 中，SM2 应为 0。
- REN：允许串行接收位，由软件置位或清零。REN = 1 时，允许接收；REN = 0 时，禁止接收。
- TB8：发送数据的第 9 位。在方式 2 和方式 3 中，由软件置位或复位，可做奇偶校验位。在多机通信中，可作为区别地址帧或数据帧的标识位，一般约定地址帧时 TB8 为 1，数据帧时 TB8 为 0。
- RB8：接收数据的第 9 位，功能同 TB8。
- TI：发送中断标志位。在方式 0 中，发送完 8 位数据后，由硬件置位；在其他方式中，在发送停止位之初由硬件置位。因此 TI 是发送完一帧数据的标志，可以用指令 JBC TI, rel 来查询是否发送结束。TI = 1 时，也可向 CPU 申请中断，响应中断后必须由软件清除 TI。
- RI：接收中断标志位。在方式 0 中，接收完 8 位数据后，由硬件置位；在其他方式中，在接收停止位的中间由硬件置位。与 TI 一样，也可以通过 JBC RI, rel 来查询是否接收完一帧数据。RI = 1 时，也可申请中断，响应中断后必须由软件清除 RI。

3) 电源及波特率选择寄存器 PCON PCON 主要是为 CHMOS 型单片机的电源控制而设置的专用寄存器，不可以位寻址，字节地址为 87H。在 HMOS 的 8051 单片机中，PCON 除了最高位以外其他位都是虚设的。其格式如表 3-14 所示。

表 3-14 PCON 的各位定义

SMOD	×	×	×	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

与串行通信有关的只有 SMOD 位，SMOD 为波特率选择位。在方式 1、2、3 中，串行通信的波特率与 SMOD 有关。当 SMOD = 1 时，通信波特率乘以 2，当 SMOD = 0 时，波特率不变。

其他各位用于电源管理，在此不再赘述。

### 3. 串行口的工作方式

MCS-51 的串行口有 4 种工作方式，由 SCON 中的 SM1、SM0 位决定，其编码和功能如表 3-15 所示。



表 3-15 串行口工作方式

SM0	SM1	方 式	功 能 说 明	波 特 率
0	0	方式 0	移位寄存器方式	$f_{osc}/12$
0	1	方式 1	10 位 UART	可变
1	0	方式 2	11 位 UART	$f_{osc}/64$ 或 $f_{osc}/32$
1	1	方式 3	11 位 UART	可变

方式 0 和方式 2 的波特率是固定的，而方式 1 和方式 3 的波特率是可变的，由 T1 的溢出率决定。

**1) 方式 0** 在方式 0 下，串行口作同步移位寄存器用，其波特率固定为  $f_{osc}/12$ 。串行数据从 RXD (P3.0) 端输入或输出，同步移位脉冲由 TXD (P3.1) 送出。这种方式常用于扩展 I/O 口。

(1) 发送。当一个数据写入串行口发送缓冲器 SBUF 时，串行口将 8 位数据以  $f_{osc}/12$  的波特率从 RXD 引脚输出（低位在前），发送完置中断标志 TI 为 1，请求中断。在再次发送数据之前，必须由软件清 TI 为 0。具体接线图如图 3-15 所示，其中 74LS164 为串入并出移位寄存器。

(2) 接收。在满足  $REN=1$  和  $RI=0$  的条件下，串行口即开始从 RXD 端以  $f_{osc}/12$  的波特率输入数据（低位在前），当接收完 8 位数据后，置中断标志 RI 为 1，请求中断。在再次接收数据之前，必须由软件清 RI 为 0。具体接线图如图 3-16 所示，其中 74LS165 为并入串出移位寄存器。

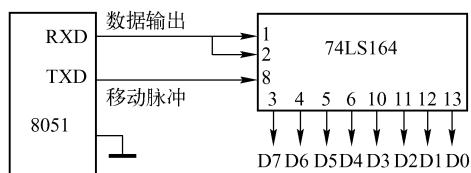


图 3-15 方式 0 用于扩展 I/O 口输出

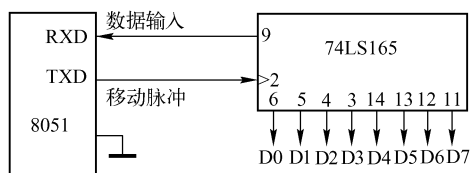


图 3-16 方式 0 用于扩展 I/O 口输入

串行控制寄存器 SCON 中的 TB8 和 RB8 在方式 0 中未用。值得注意的是，每当发送或接收完 8 位数据后，硬件会自动置 TI 或 RI 为 1，CPU 响应 TI 或 RI 中断后，必须由用户用软件清 0。使用方式 0 时，SM2 必须为 0。关于串行口方式 0 在扩展 I/O 方面的应用，软件编程本书不再做介绍。

**2) 方式 1** 方式 1 是 10 位为一帧的异步串行通信方式。发送或接收一帧信息，包括 1 位起始位 0、8 位数据位和 1 位停止位 1。其帧格式如图 3-17 所示。

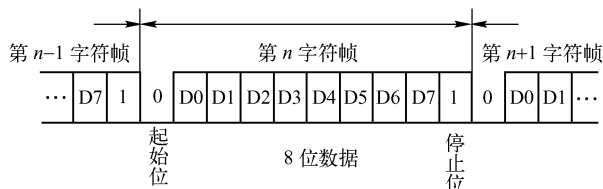


图 3-17 10 位的帧格式

(1) 发送。发送时,数据从 TXD 输出,当数据写入发送缓冲器 SBUF 后,启动发送器发送。当发送完一帧数据后,置中断标志 TI 为 1。方式 1 所传送的波特率取决于定时器 T1 的溢出率和 PCON 中的 SMOD 位。

(2) 接收。接收时,由 REN 置 1 允许接收,串行口采样 RXD,当采样 1 到 0 的跳变时,确认是起始位 0,就开始接收一帧数据。当 RI = 0 且停止位为 1 或 SM2 = 0 时,停止位进入 RB8 位,同时置中断标志 RI,否则信息将丢失。所以,方式 1 接收时,应先用软件清除 RI 或 SM2 标志。

3) 方式 2 在方式 2 中,串行口为 11 位 UART,传送波特率与 SMOD 有关。发送或接收一帧数据包括 1 位起始位 0、8 位数据位、1 位可编程位(用于奇偶校验)和 1 位停止位 1。其帧格式如图 3-18 所示。

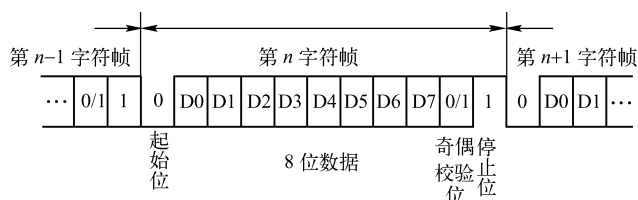


图 3-18 11 位的帧格式

(1) 发送。发送时,先根据通信协议由软件设置 TB8,然后用指令将要发送的数据写入 SBUF,启动发送器。写 SBUF 的指令,除了将 8 位数据送入 SBUF 外,同时还将 TB8 装入发送移位寄存器的第 9 位,并通知发送控制器进行一次发送。一帧信息从 TXD 发送,在送完一帧信息后, TI 被自动置 1,在发送下一帧信息之前, TI 必须由中断服务程序或查询程序清 0。

(2) 接收。当 REN = 1 时,允许串行口接收数据。数据由 RXD 端输入,接收 11 位的信息。当接收器采样到 RXD 端的负跳变并判断起始位有效后,开始接收一帧信息。当接收器接收到第 9 位数据后,若同时满足两个条件 RI = 0; SM2 = 0 或接收到的第 9 位数据为 1,则接收数据有效,8 位数据送入 SBUF,第 9 位送入 RB8,并置 RI = 1。若不满足上述两个条件,则信息丢失。

4) 方式 3 方式 3 为波特率可变的 11 位 UART 通信方式,除了波特率以外,方式 3 和方式 2 完全相同。

#### 4. 波特率设计

在串行通信中,收发双方对传送的数据速率即波特率要有一定的约定。通过前面的论述我们已经知道, MCS-51 单片机的串行口通过编程有 4 种工作方式。其中方式 0 和方式 2 的波特率是固定的,方式 1 和方式 3 的波特率可变,由定时器 T1 的溢出率决定,下面具体分析。

1) 方式 0 和方式 2 在方式 0 中,波特率为时钟频率的 1/12,即  $f_{\text{osc}}/12$ ,固定不变。

在方式 2 中,波特率取决于 PCON 中的 SMOD 值,当 SMOD = 0 时,波特率为  $f_{\text{osc}}/64$ ;当 SMOD = 1 时,波特率为  $f_{\text{osc}}/32$ ,即波特率  $= \frac{2^{\text{SMOD}}}{64} \cdot f_{\text{osc}}$ 。

2) 方式 1 和方式 3 在方式 1 和方式 3 下,波特率由定时器 T1 的溢出率和 SMOD 共同决定。即方式 1 和方式 3 的波特率  $= \frac{2^{\text{SMOD}}}{32} \cdot \text{T1 的溢出率}$ 。

其中 T1 的溢出率取决于单片机定时器 T1 的计数速率和定时器的预置值。计数速率与 TMOD 寄存器中的 C/T 位有关, 当  $C/\bar{T}=0$  时, 计数速率为  $f_{osc}/12$ , 当  $C/\bar{T}=1$  时, 计数速率为外部输入时钟频率。

实际上, 当定时器 T1 作波特率发生器使用时, 通常是工作在模式 2, 即自动重装载的 8 位定时器, 此时 TL1 作计数用, 自动重装载的值在 TH1 内。设计数的预置值 (初始值) 为 X, 则每过  $256-X$  个机器周期, 定时器溢出一次。为了避免溢出而产生不必要的中断, 此时应禁止 T1 中断。溢出周期为:

$$\frac{12}{f_{osc}} \cdot (256 - X)$$

溢出率为溢出周期的倒数, 所以:

$$\text{波特率} = \frac{2^{SMOD}}{32} \cdot \frac{f_{osc}}{12(256 - X)}$$

表 3-16 列出了晶振为 11.059MHz 时各种常用的波特率。

表 3-16 定时器 T1 产生的常用波特率

波 特 率	$f_{osc}$	SMOD	定时器 T1		
			C/ $\bar{T}$	模式	初始值
方式 0: 1MHz	12MHz	×	×	×	×
方式 2: 375kHz	12MHz	1	×	×	×
方式 1、3: 62.5kHz	12MHz	1	0	2	FFH
19.2kHz	11.059MHz	1	0	2	FDH
9.6kHz	11.059MHz	0	0	2	FDH
4.8kHz	11.059MHz	0	0	2	FAH
2.4kHz	11.059MHz	0	0	2	F4H
1.2kHz	11.059MHz	0	0	2	E8H
137.5kHz	11.986MHz	0	0	2	1DH
110Hz	6MHz	0	0	2	72H
110Hz	12Hz	0	0	1	FEEDH



### 3.4 51 单片机的定时/计数器

定时/计数器是单片机系统一个重要的部件, 其工作方式灵活、编程简单、使用方便, 可用来实现定时控制、延时、频率测量、脉宽测量、信号发生、信号检测等。此外, 定时/计数器还可作为串行通信中的波特率发生器。

#### ► 基本概念

- ☆ 计数: 计数是指对外部事件的个数进行计量, 其实质就是对外部输入脉冲的个数进行计量。实现计数功能的器件称为计数器。
- ☆ 定时: 8051 单片机中的定时器和计数器是一个部件, 只不过计数器记录的是外界发生的事件, 而定时器则是由单片机内部提供一个非常稳定的计数源进行计时的。这个计数源是由单片机的晶振经过 12 分频后获得的一个脉冲源。所以定时器计数脉冲的时间间隔与晶振有关。

- 定时的种类
- ☆ 软件定时：利用执行一个循环程序进行时间延迟。其特点是定时时间精确，不需外加硬件电路，但占用 CPU 时间。因此软件定时的时间不宜过长。
  - ☆ 硬件定时：利用硬件电路实现定时。其特点是不占用 CPU 时间，通过改变电路元器件参数来调节定时，但使用不够灵活方便。对于时间较长的定时，常用硬件电路来实现。
  - ☆ 可编程定时器：通过专用的定时/计数器芯片实现。其特点是通过对系统时钟脉冲进行计数实现定时，定时时间可通过程序设定的方法改变，使用灵活方便。也可实现对外部脉冲的计数功能。
- MCS-51 内部定时/计数器。MCS-51 单片机内部有两个 16 位可编程的定时/计数器，简称 T0 和 T1，可作为定时器也可作为计数器，它们均是二进制加法计数器，当计数器计满回零时能自动产生溢出中断请求，表示定时时间已到或计数已终止。适用于定时控制、延时、外部计数和检测等。
- ☆ 计数器：对引脚 T0（P3.4）和 T1（P3.5）输入的外部脉冲信号计数，当输入脉冲信号为从 1 到 0 的负跳变时，计数器自动加 1。计数的最高频率一般为振荡频率的 1/24。
  - ☆ 定时器：对系统晶振振荡脉冲的 12 分频输出进行计数。

1. 定时/计数器的基本结构

定时/计数器的基本结构如图 3-19 所示。T0 由 TH0 和 TL0 两个 8 位二进制加法计数器组成 16 位二进制加法计数器；T1 由 TH1 和 TL1 两个 8 位二进制加法计数器组成 16 位二进制加法计数器。

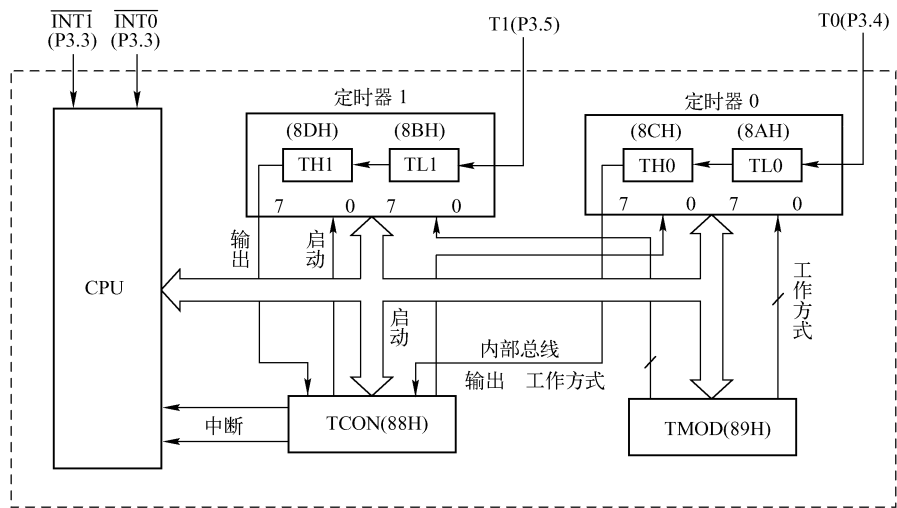


图 3-19 定时/计数器的基本结构

1) 定时器方式控制寄存器 TMOD 定时器方式控制寄存器 TMOD 的地址为 89H，不可位寻址。TMOD 寄存器中的高 4 位定义 T1，低 4 位定义 T0，如表 3-17 所示。其中 M1、M0 用来确定所选工作方式，如表 3-18 所示。

表 3-17 TMOD 各位的定义

位序	D7	D6	D5	D4	D3	D2	D1	D0
位符号	GATE	C/T	M1	M0	GATE	C/T	M1	M0
定时/计数器 T1				定时/计数器 T0				

表 3-18 TMOD 控制位功能

符 号	功 能 说 明
GATE	门控位 GATE = 0, 用运行控制位 TR0 (TR1) 启动定时器 GATE = 1, 用外中断请求信号输入端 (INT1 或 INT0) 和 TR0 (TR1) 共同启动定时器
C/ $\bar{T}$	定时方式或计数方式选择位 C/ $\bar{T}$ = 0, 定时工作方式 C/ $\bar{T}$ = 1, 计数工作方式
M1 M0	工作方式选择位 M1 M0 = 00, 方式 0, 13 位计数器 M1 M0 = 01, 方式 1, 16 位计数器 M1 M0 = 10, 方式 2, 具有自动再装入的 8 位计数器 M1 M0 = 11, 方式 3, 定时器 0 分成两个 8 位计数器, 定时器 1 停止计数

2) 定时器控制寄存器 TCON 定时器控制寄存器 TCON 的地址为 88H, 可以位寻址, 各位的定义见表 3-19。TCON 主要用于控制定时器的操作及中断控制。有关中断内容在 3.2 节 51 单片机的中断系统中已说明, 此处只对定时控制功能加以介绍, 如表 3-20 所示。

表 3-19 TCON 各位的定义

位 地 址	8F	8E	8D	8C	8B	8A	89	88
位符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

表 3-20 TCON 控制位功能

符 号	功 能 说 明
TF1	计数/计时 1 溢出标志位。计数/计时 1 溢出 (计满) 时, 该位置 1。在中断方式时, 此位作中断标志位, 在转向中断服务程序时由硬件自动清 0。在查询方式时, 也可以由程序查询和清 0
TR1	定时/计数器 1 运行控制位 TR1 = 0, 停止定时/计数器 1 工作 TR1 = 1, 启动定时/计数器 1 工作 该位由软件置位和复位
TF0	计数/计时 0 溢出标志位。计数/计时 0 溢出 (计满) 时, 该位置 1。在中断方式时, 此位作中断标志位, 在转向中断服务程序时由硬件自动清 0。在查询方式时, 也可以由程序查询和清 0
TR0	定时/计数器 0 运行控制位 TR0 = 0, 停止定时/计数器 0 工作 TR0 = 1, 启动定时/计数器 0 工作 该位由软件置位和复位

系统复位时, TMOD 和 TCON 寄存器的每一位都清零。

## 2. 定时/计数器的 4 种工作模式

用户可通过编程对专用寄存器 TMOD 中的 M1、M0 位进行设置, 选择 4 种操作方式。

1) 方式 0 (以 T0 为例) 在此方式中, 定时寄存器由 TH0 的 8 位和 TL0 的 5 位 (其余位不用) 组成一个 13 位计数器。当 GATE = 0 时, 只要 TCON 中的 TR0 为 1, 13 位计数器就开始计数; 当 GATE = 1 及 TR0 = 1 时, 13 位计数器是否计数取决于 INT0 引脚信号, 当 INT0 由 0 变为 1 时开始计数, 当 INT0 由 1 变为 0 时停止计数。

当 13 位计数器溢出时, TCON 的 TF0 位就由硬件置 1, 同时将计数器清 0。

当方式 0 为定时工作方式时，定时时间计算公式为：

$$(2^{13} - \text{计数初值}) \times \text{晶振周期} \times 12$$

当方式 0 为计数工作方式时，计数值的范围是：1 ~  $2^{13}$  (8 192)。

方式 0 的内部逻辑结构图如图 3-20 所示。

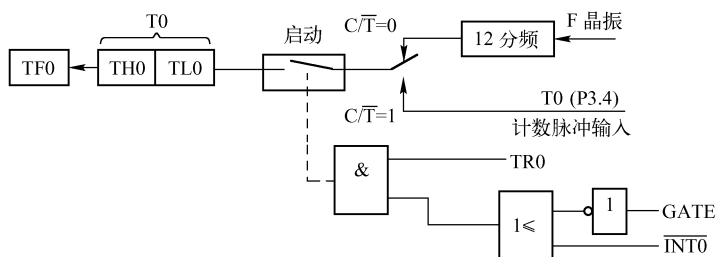


图 3-20 方式 0 的内部逻辑结构图

2) 方式 1 方式 1 采用 16 位计数结构的工作方式，其余与方式 0 相同。显然方式 1 的定时时间计算公式为：

$$(2^{16} - \text{计数初值}) \times \text{晶振周期} \times 12$$

计数范围是：1 ~  $2^{16}$  (65 536)。

3) 方式 2 方式 2 是由 TL 组成 8 位计数器，TH 作为常数缓冲器，由软件预置初始值。当 TL 产生溢出时，一方面使溢出标志位 TF 置 1；同时把 TH 的 8 位数据重新装入 TL 中，即方式 2 具有自动重新加载功能。方式 2 的内部逻辑结构如图 3-21 所示（以定时/计数器 0 为例）。

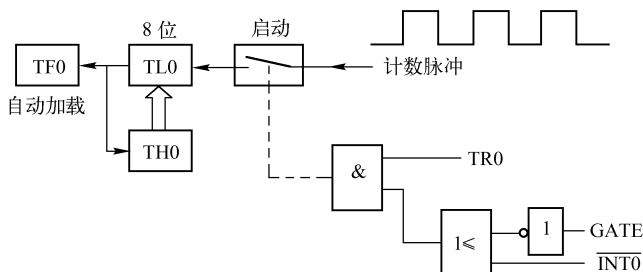


图 3-21 方式 2 内部逻辑结构图

4) 方式 3 在方式 3 中，TL0 和 TH0 成为两个相互独立的 8 位计数器。TL0 占用了全部 T0 的控制位和信号引脚，即 GATE、C/T、TR0、TF0 等，而 TH0 只作定时器使用。而且由于定时/计数器 0 的控制位已被 TL0 独占，因此 TH0 只好借用定时/计数器 1 的控制位 TR1 和 TF1 进行工作。

同时，由于 TR1、TF1 已“出借”给 TH0，TH1 和 TL1 的溢出就送给串行口，作为串行口时钟信号发生器（即波特率信号发生器），并且只要设置好工作方式（方式 0、方式 1、方式 2）及计数初值，T1 无须启动便可自动运行。若要 T1 停止工作，只要将其设置为工作方式 3 即可。

### 3. 定时/计数器的编程与应用

定时/计数器可按片内机器周期定时，也可对由 T0/T1 引脚输入的一个负脉冲进行加法

计数。在应用时,其工作方式和工作过程均可通过程序设定和控制,因此,定时/计数器在工作前必须先对其进行初始化,计算和设置初值。

#### 1) 定时器 T0/T1 中断申请过程

(1) 在已经开放 T0/T1 中断允许且已被启动的前提下, T0/T1 加 1 计满溢出时 TF0/TF1 标志位自动置 1;

(2) CPU 检测到 TCON 中 TF0/TF1 变 1 后,将产生指令 LCALL 000BH/LCALL 001BH,执行中断服务程序;

(3) TF0/TF1 标志位由硬件自动清 0,以备下次中断申请。

#### 2) 定时/计数器初始化的步骤

(1) 写 TMOD,设置定时/计数器的工作方式;

(2) 计算定时/计数器的初值,写入 TH0/TH1、TL0/TL1;

(3) 设置 IE、IP,以开放相应的中断和设定中断优先级。

#### 3) 定时/计数器的定时/计数范围

(1) 工作方式 0: 13 位定时/计数器方式,最大计数值  $= 2^{13} = 8192$ 。

(2) 工作方式 1: 16 位定时/计数器方式,最大计数值  $= 2^{16} = 65536$ 。

(3) 工作方式 2 和工作方式 3: 8 位定时/计数器方式,最大计数值  $= 2^8 = 256$ 。

#### 4) 计数器初值的计算

方法:用最大计数量减去需要的计数次数。即

$$T_c = M - C$$

式中  $T_c$ ——计数器需要预置的初值。

$M$ ——计数器的模值(最大计数值)。方式 0 时,  $M = 2^{13}$ ; 方式 1 时,  $M = 2^{16}$ ; 方式 2、3 时,  $M = 2^8$ 。

$C$ ——计数器计满回 0 所需的计数值,即设计任务要求的计数值。

例如:啤酒流水线上一个包装是 12 瓶,要求每到 12 瓶就产生一个动作。用单片机的工作方式 0 来控制,则应当预置的初值为:

$$T_c = M - C = 2^{13} - 12 = 8180$$

#### 5) 定时器初值的计算 定时时间的计算公式为:

$$T = (M - T_c) \times T_0 \text{ (或 } T_c = M - T/T_0)$$

式中  $T$ ——定时器的定时时间,即设计任务要求的定时时间;

$T_0$ ——计数器计数脉冲的周期,即单片机系统主频周期的 12 倍;

$M$ ——计数器的模值;

$T_c$ ——定时器需要预置的初值。

若设初值  $T_c = 0$ ,则定时器定时时间为最大。若设单片机系统主频为 12MHz,则各种工作方式定时器的最大定时时间为:

工作方式 0,  $T_{\max} = 2^{13} \times 1\mu s = 8.192ms$ ;

工作方式 1,  $T_{\max} = 2^{16} \times 1\mu s = 65.536ms$ ;

工作方式 2 和 3,  $T_{\max} = 2^8 \times 1\mu s = 0.256ms$ 。



### 3.5 思考与练习

1. MCS-51 单片机的芯片包含哪些主要逻辑功能部件？各有什么主要功能？
2. MCS-51 单片机的 EA 信号有什么功能？在使用 8031 时 EA 信号引脚应如何处理？
3. 简述程序状态字 PSW 中各个位的作用。
4. 内部 RAM 低 128 单元划分为哪 3 个主要部分？说明各部分的使用特点。
5. 堆栈有哪些功能？堆栈指示器（SP）的作用是什么？在程序设计时，为什么要对 SP 重新赋值？
6. MCS-51 的 4 个 I/O 口在使用上有哪些分工和特点？
7. MCS-51 单片机的片内、片外存储器如何选择？
8. 什么是指令周期、机器周期和时钟周期？如何计算机器周期的确切时间？
9. 使单片机复位有哪几种方法？复位后机器的初始状态如何？
10. MCS-51 单片机只有两个外部中断，如果需要单片机处理两个以上的外部中断，该如何进行？
11. 如果要开放外部中断 0 和串口的中断，而屏蔽其他中断，控制字是什么？如何实现这个控制结果？
12. MCS-51 单片机的中断系统有几个中断源？分别是什么？事先约定的优先顺序是怎样的？
13. 各中断源对应的中断服务程序的入口地址是否能任意设定？
14. 如果想将中断服务程序放置在程序存储区的任意区域，在程序中应该作何种设置？请举例加以说明。
15. 什么是串行异步通信？有哪几种帧格式？
16. 设计并编程，完成单片机的双机通信程序，将甲机片外 RAM 1000H ~ 100FH 的数据块通过串行口传送到乙机的 30H ~ 3FH 单元。
17. 设单片机的晶振频率为 6MHz，编程，用定时器 T0 在 P1.0 口连续输出频率为 100Hz 的方波。
18. 用 T0 对外部脉冲进行计数，每计满 50 次后从 P1.0 输出宽度为 1ms 的高电平，如此循环下去（时钟频率为 12MHz）。



# 第 4 章    PROTEUS 电路设计与仿真基础

电路仿真技术就是通过软件来实现并检验所设计电路功能的过程。PROTEUS 不仅是模拟电路、数字电路、模数混合电路的设计平台，更是目前世界上最先进的单片机和嵌入式系统的设计与仿真平台。它除了具有和其他 EDA 工具软件一样的原理图绘制、PCB 自动或人工布线及电路仿真功能外，其电路仿真还具有互动性，针对微处理器的应用，还可以直接在基于原理图的虚拟模型上进行编程，并实现软件源码级的实时仿真调试，若有显示及输出，还能看到运行后输入/输出的效果。它真正实现了在计算机上完成从原理图设计、电路分析与仿真、单片机代码级调试与仿真、系统测试与功能验证到形成 PCB 的完整的电子线路设计、功能仿真的整个过程。



## 4.1    PROTEUS 软件简介

PROTEUS 是英国 Labcenter 公司开发的电路分析与仿真软件。该软件的特点是：① 集原理图设计、仿真和 PCB 设计于一体，真正实现从概念到产品的完整电子设计工具；② 具有模拟电路、数字电路、单片机应用系统、嵌入式系统（不高于 ARM7）设计与仿真功能；③ 具有全速、单步、设置断点等多种形式的调试功能；④ 具有各种信号源和电路分析所需的虚拟仪表；⑤ 支持 Keil C51、 $\mu$ Vision2、MPLAB 等第三方的软件编译和调试环境；⑥ 具有强大的原理图到 PCB 的设计功能，可以输出多种格式的电路设计报表。拥有 PROTEUS 电子设计工具，就相当于拥有了一个电子设计和分析平台。

### 1. PROTEUS 软件组成

PROTEUS 软件自 1989 年问世至今，经历了 20 多年的发展历史，功能得到了不断的完善，性能越来越好，全球的用户也越来越多。PROTEUS 之所以在全球得到应用，原因是它具有自身的特点和结构。PROTEUS 电子设计软件由原理图输入模块（简称 ISIS）、混合模型仿真器、动态器件库、高级图形分析模块、处理器仿真模型及 PCB 设计编辑（简称 ARES）6 部分组成，如图 4-1 所示。

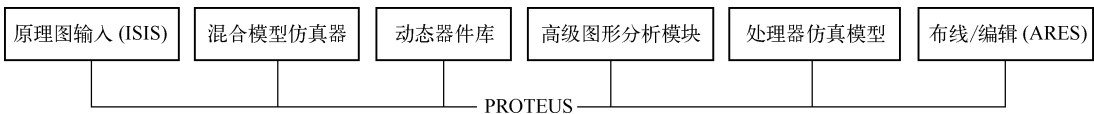


图 4-1    PROTEUS 基本组成

### 2. PROTEUS 软件资源

**1) 工具** 工具包括标准工具和绘图工具，标准工具的内容与菜单栏的内容一一对应，绘图工具栏有丰富的操作工具，选择不同的按钮会得到不同的工具。

### (1) 操作工具：PROTEUS 提供下列操作工具。

- Component：选择元器件。
- Junction dot：在原理图中标注连接点。
- Wire label：标注网络标号。
- Text script：在电路中输入说明文本。
- Bus：绘制总线。
- Bus-circuit：绘制子电路块。
- Instant edit mode：选择元器件。
- Inter-sheet terminal：对象选择器列出输入/输出、电源、地等终端。
- Device Pin：对象选择器列出普通引脚、时钟引脚、反电压引脚和短接引脚等。
- Simulation graph：对象选择器列出各种仿真分析所需的图表。
- Tape recorder：当对设计电路分割仿真时采用此模式。
- Generator：对象选择器列出各种激励源。
- Voltage probe：电压探针，电路进入仿真模式时可显示各探针处的电压值。
- Current probe：电流探针，电路进入仿真模式时可显示各探针处的电流值。
- Virtual instrument：对象选择器列出各种虚拟仪器。

### (2) 图形绘制工具：如下所示。

- 2D graphics line：绘制直线（用于创建元器件或表示图表时绘制线）。
- 2D graphics box：绘制方框。
- 2D graphics circle：绘制圆。
- 2D graphics arc：绘制弧。
- 2D graphics path：绘制任意形状图形。
- 2D graphics text：文本编辑，用于插入说明。
- 2D graphics symbol：用于选择各种符号元器件。
- Makers for component origin etc：用于产生各种标记图标。
- Set rotation：方向旋转按钮，以  $90^\circ$  偏置改变元器件的放置方向。
- Horizontal reflection：水平镜像旋转按钮。
- Vertical reflection：垂直镜像旋转按钮。

## 2) 虚拟仪器

(1) 电路激励源：PROTEUS 中提供了 13 种信号源，对于每一种信号源参数又可进行设置。

- DC：直流电压源。
- Sine：正弦波发生器。
- Pulse：脉冲发生器。
- Exp：指数脉冲发生器。
- SFFM：单频率调频波信号发生器。
- Pwlin：任意分段线性脉冲信号发生器。
- File：File 信号发生器，数据来源于 ASCII 文件。
- Audio：音频信号发生器，数据来源于 wav 文件。
- DState：稳态逻辑电平发生器。
- DEdge：单边沿信号发生器。
- DPulse：单周期数字脉冲发生器。
- DClock：数字时钟信号发生器。

➤ DPattern: 模式信号发生器。

(2) 电路功能分析: PROTEUS 中提供了 9 种电路分析工具, 在进行电路设计时, 用来测试电路的工作状态。

- 虚拟示波器 (VIRTUAL; OSCILLOSCOPE)
- 逻辑分析仪 (LOGIC ANALYSER)
- 计数/定时器 (COUNTER TIMER)
- 虚拟终端 (VIRTUAL TERMINAL)
- 信号发生器 (SIGNAL GENERATOR)
- 模式发生器 (PATTERN GENERATOR)
- 交直流电压表和电流表 (AC/DC voltmeters/ammeters)
- SPI 调试器 (SPI DEBUGGER)
- I<sup>2</sup>C 调试器 (I<sup>2</sup>C DEBUGGER)

(3) 电路图表分析: PROTEUS 中提供了 13 种分析图表, 在进行电路高级仿真时, 用来精确分析电路的技术指标。

- 模拟图表 (ANALOGUE)
- 数字图表 (DIGITAL)
- 混合分析图表 (MIXED)
- 频率分析图表 (FREQUENCY)
- 转移特性分析图表 (TRANSFER)
- 噪声分析图表 (NOISE)
- 失真分析图表 (DISTORTION)
- 傅里叶分析图表 (FOURIER)
- 音频分析图表 (AUDIO)
- 交互分析图表 (INTERACTIVE)
- 一致性分析图表 (CONFORMANCE)
- 直流扫描分析图表 (DC SWEEP)
- 交流扫描分析图表 (AC SWEEP)

(4) 测试探针: PROTEUS 中提供了电流和电压探针, 用来测试电路中的电流和电压值。值得注意的是, 电流探针的方向一定要与电路的导线平行。

- 电压探针 (Voltage probes): 既可在模拟仿真中使用, 也可在数字仿真中使用。在模拟电路中记录真实的电压值, 而在数字电路中记录逻辑电平及其强度。
- 电流探针 (Current probes): 仅在模拟电路仿真中使用, 可显示电流方向和电流瞬时值。

**3) 元件** PROTEUS 提供了大量元器件的原理图符号和 PCB 封装, 在绘制原理图之前必须知道每个元器件对应的库, 在自动布线之前必须知道对应元器件的封装, 下面是常用的元器件库。

#### (1) 元件库

- Device. LIB (电阻、电容、二极管、三极管等常用元件库)
- Active. LIB (活性元器件库)
- Diode. LIB (二极管和整流桥库)
- Display. LIB (LED 和 LCD 显示器件库)
- Bipolar. LIB (三极管库)
- Fet. LIB (场效应管库)

- Realtime. LIB (实时元器件库)
- Asimmdl. LIB (常用的模拟器件库)
- Dsimmdl. LIB (数字器件库)
- Valves. LIB (电子管库)
- 74STD. LIB (74 系列标准 TTL 元器件)
- 74AS. LIB (74 系列 AS TTL 元器件)
- 74LS. LIB (74 系列 LS TTL 元器件)
- 74ALS. LIB (74 系列 ALS TTL 元器件库)
- 74S. LIB (74 系列肖特基 TTL 元器件库)
- 74F. LIB (74 系列快速 TTL 元器件库)
- 74HC. LIB (74 系列和 4000 系列高速 CMOS 元器件)
- ANALOG. LIB (调节器、运放和数据采样 IC)
- CAPACITORS. LIB (电容)
- CMOS. LIB (4000 系列 CMOS 元器件)
- ECL. LIB (ECL 10000 系列元器件)
- I<sup>2</sup>C MEM. LIB (I<sup>2</sup>C 存储器)
- MEMORY. LIB (存储器)
- MICRO. LIB (常用微处理器)
- OPAMP. LIB (运算放大器)
- RESISTORS. LIB (电阻)

## (2) 封装库

- PACKAGE. LIB (二极管、三极管、IC、LED 等常用元件封装库)
- SMTDISC. LIB (常用元件的表贴封装库)
- SMTCHIP. LIB (LCC、PLCC、CLCC 等器件封装)
- SMTBGA. LIB (常用接插件封装)

## 3. 基于 PROTEUS 的产品设计方法

基于 PROTEUS 的电子产品开发流程如图 4-2 所示。

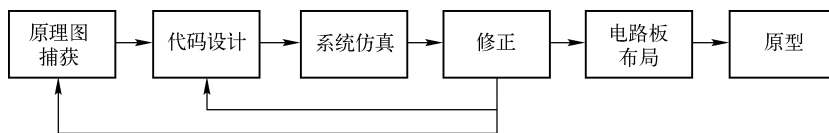


图 4-2 基于 PROTEUS 的电子产品开发流程

基于 PROTEUS 的电子产品开发优点如下：

- 只要完成原理图设计就可用于系统的测试。
- PROTEUS 的交互仿真特性使软件的调试和测试在布板之前完成。
- 硬件设计的改动容易得如软件设计改动一样。

## 4. PROTEUS 软件的应用

1) PROTEUS 适合教学 PROTEUS 是一个巨大的教学资源，可以用于：

- 模拟电路与数字电路的教学与实验；

- 单片机和嵌入系统软件的教学与实验；
- 微控制器系统的综合实验；
- 创新实验与毕业设计；
- 项目设计与产品开发。

## 2) PROTEUS 适合技能考评与电子竞赛

- 单片机技能考评的必要性与难点；
- PROTEUS 能提供考试所需的所有资源；
- PROTEUS 能直观评估硬件电路的设计正确性；
- PROTEUS 能通过硬件原理图直观调试软件；
- PROTEUS 能验证整个设计的功能；
- 测试可控、易评估、易实施；
- 有广泛的使用案例。

## 3) PROTEUS VSM 适合产品开发

- PROTEUS 软件提供了从产品概念到设计完成的完整仿真与开发平台。
- 具有仿真与分析功能，可将产品开发中存在的问题消灭在萌芽中，从而减少开发风险。
- 软、硬件的交互仿真与测试大大减少了后期测试工作量。
- 便于项目管理与团队开发。



# 4.2 PROTEUS 软件的基本操作

## 1. PROTEUS ISIS 操作界面

PROTEUS ISIS 的工作界面是一种标准的 Windows 界面，如图 4-3 所示，包括标题栏、主菜单、标准工具栏、绘图工具栏、状态栏、对象选择按钮、预览对象方位控制按钮、仿真进程控制按钮、预览窗口、对象选择器窗口、图形编辑窗口。

为了方便介绍，分别对窗口内各部分进行中文说明（见图 4-3）。下面简单介绍各部分的功能。

**1) 原理图编辑窗口 (The Editing Window)** 顾名思义，该窗口是用来绘制原理图的。蓝色方框内为可编辑区，元件要放到它里面。注意，这个窗口是没有滚动条的，可以用预览窗口来改变原理图的可视范围。同时，它的操作是不同于常用的 Windows 应用程序的，正确的操作是：中键放/缩原理图；左键放置元件；右键选择元件；双击右键删除元件；先右键后左键编辑元件属性；先右键后左键拖动元件；连线用左键，删除用右键。

**2) 预览窗口 (The Overview Window)** 该窗口可显示两个内容，一个是，当在元件列表中选择一元件时，它会显示该元件的预览图；另一个是，当鼠标焦点落在原理图编辑窗口时（即放置元件到原理图编辑窗口后或在原理图编辑窗口中单击鼠标后），它会显示整张原理图的缩略图，并会显示一个绿色的方框，绿色方框里面的内容就是当前原理图窗口中显示的内容。因此，可用鼠标在它上面单击来改变绿色方框的位置，从而改变原理图的可视范围。

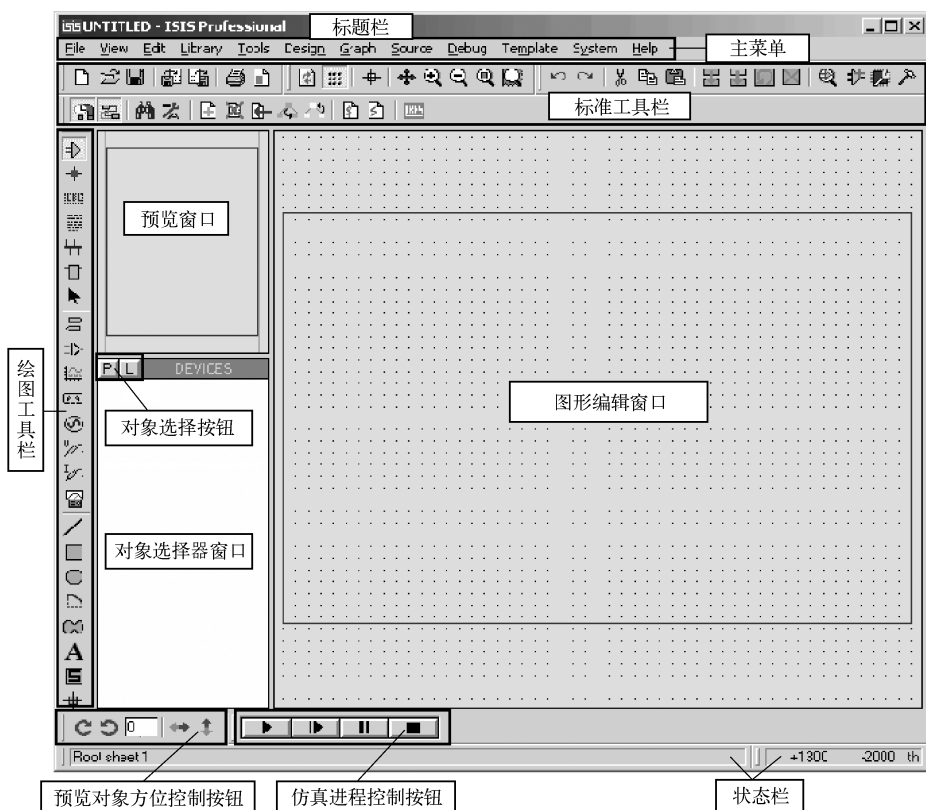


图 4-3 PROTEUS ISIS 工作界面

### 3) 模型选择工具栏 (Mode Selector Toolbar)

(1) 主要模型 (Main Modes):

: 选择元件 (components) (默认选择的)。

: 放置连接点。

: 放置标签 (用总线时会用到)。

: 放置文本。

: 用于绘制总线。

: 用于放置子电路。

: 用于即时编辑元件参数 (先单击该图标再单击要修改的元件)。

(2) 配件 (Gadgets):

: 终端接口 (terminals), 有 VCC、地、输出、输入等接口。

: 器件引脚, 用于绘制各种引脚。

: 仿真图表 (graph), 用于各种分析, 如 Noise Analysis。

: 录音机。









: 信号发生器 (generators)。

: 电压探针, 使用仿真图表时要用到。

: 电流探针, 使用仿真图表时要用到。

: 虚拟仪表, 有示波器等。

(3) 2D 图形 (2D Graphics):

- : 画各种直线。
- : 画各种方框。
- : 画各种圆。
- : 画各种圆弧。
- : 画各种多边形。
- : 画各种文本。
- : 画符号。
- : 画原点等。

4) 元件列表 (The Object Selector) 用于挑选元件 (components)、终端接口 (terminals)、信号发生器 (generators)、仿真图表 (graph) 等。例如, 选择“元件 (components)”, 单击“P”按钮会打开挑选元件对话框, 选择一个元件后 (单击“OK”按钮后), 该元件会在元件列表中显示, 以后要用到该元件时, 只需在元件列表中选择即可。


#### 5) 方向工具栏 (Orientation Toolbar)


旋转: 旋转角度只能是 90 的整数倍。


翻转: 完成水平翻转和垂直翻转。


使用方法: 先用右键单击元件, 再单击 (左击) 相应的旋转图标。

#### 6) 仿真工具栏仿真控制按钮

: 运行。

: 单步运行。

: 暂停。

: 停止。

## 2. PROTEUS 软件的基本操作

1) 绘制原理图 绘制原理图要在原理图编辑窗口中的蓝色方框内完成。原理图编辑窗口的操作是不同于常用的 Windows 应用程序的, 正确的操作是: 用左键放置元件; 右键选择元件; 双击右键删除元件; 右键拖选多个元件; 先右键后左键编辑元件属性; 先右键后左键拖动元件; 连线用左键, 删除用右键; 改连接线时, 先右击连线, 再用左键拖动; 中键放/缩原理图。具体操作见下面例子。

2) 定制自己的元件 有三个实现途径, 一是用 PROTEUS VSM SDK 开发仿真模型, 并制作元件; 另一个是在已有的元件基础上进行改造, 比如把元件改为 bus 接口的; 还有一个是利用已制作好 (别人的) 的元件, 可以到网上下载一些新元件并把它们添加到自己的元件库里。由于笔者没有 PROTEUS VSM SDK, 所以只介绍后两个。

3) Sub-Circuits 应用 用一个子电路可以把部分电路封装起来, 这样可以节省原理图窗口的空间。



## 4.3 PROTEUS ISIS 参数设置

### 1. PROTEUS ISIS 编辑环境设置

PROTEUS ISIS 编辑环境的设置主要是指模板的选择、图纸的选择、图纸的设置和格点

的设置。绘制电路图首先要选择模板,模板控制电路图外观的信息,如图形格式、文本格式、设计颜色、线条连接点大小和图形等。然后设置图纸,如设置纸张的型号、标注的字体等。图纸的格点为放置元器件、连接线路带来很多方便。

(1) 选择模板。在 PROTEUS ISIS 主界面中进行如下的操作:

选择【Template】→【Set Design Defaults】菜单项,编辑设计的默认选项;

选择【Template】→【Set Graph Colours】菜单项,编辑图形颜色;

选择【Template】→【Set Graph Styles】菜单项,编辑图形的全局风格;

选择【Template】→【Set Text Styles】菜单项,编辑全局文本风格;

选择【Template】→【Set Graphics Text】菜单项,编辑图形字体格式;

选择【Template】→【Set Junction Dots】菜单项,弹出编辑节点对话框。

 注意:模板的改变只影响当前运行的 PROTEUS ISIS,尽管这些模板有可能在保存后被别的设计调用。

(2) 选择图纸。在 PROTEUS ISIS 主界面选择【System】→【Set Sheet Sizes】菜单项,将出现如图 4-4 所示对话框,在该对话框中用户可选择图纸的大小或自定义图纸的大小。

(3) 设置文本编辑器。在 PROTEUS ISIS 主界面中选择【System】→【Set Text Editor】菜单项,出现如图 4-5 所示对话框,在该对话框中可以对文本的字体、字形、大小、效果和颜色等进行设置。

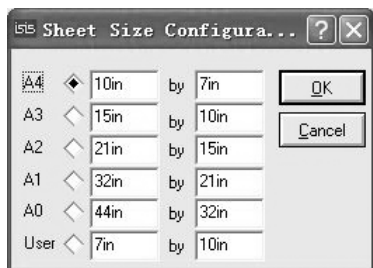


图 4-4 设置图纸大小



图 4-5 设置文本格式

(4) 设置格点。使用“View”菜单设置格点的显示或隐藏,在主界面中选择【View】→【Grid】菜单项,设置编辑窗口中的格点显示与否,使用“View”菜单设置格点的间距。选择【View】→【Snap 10th】菜单项,或 Snap 50th、Snap 0.1in、Snap 0.5in 项,可调整格点的间距(默认值为 0.1in)。

## 2. PROTEUS ISIS 系统参数设置

1) 设置 BOM 选择【System】→【Set BOM Scripts】菜单项,可打开 BOM 设置对话框,如图 4-6 所示。在 BOM 设置对话框中,可对 4 种输出格式进行设置。

单击对话框中的“Add”按钮,在“Category Heading”文本框中输入“Subcircuit”,并在“Reference(s) to match”文本框中输入“S”,单击“OK”按钮,则可将新的“Category”添加到“BOM”,如图 4-7 所示。



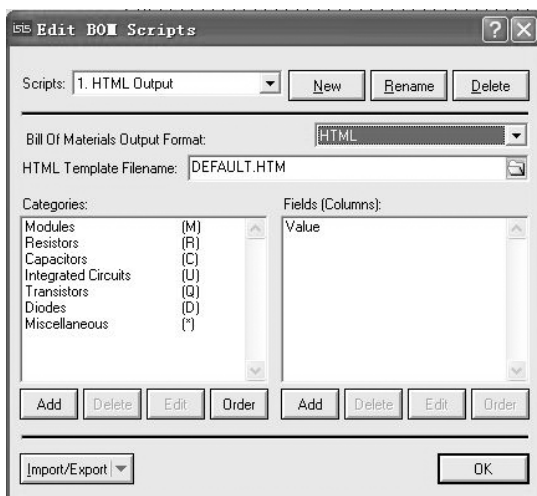


图 4-6 BOM 设置对话框

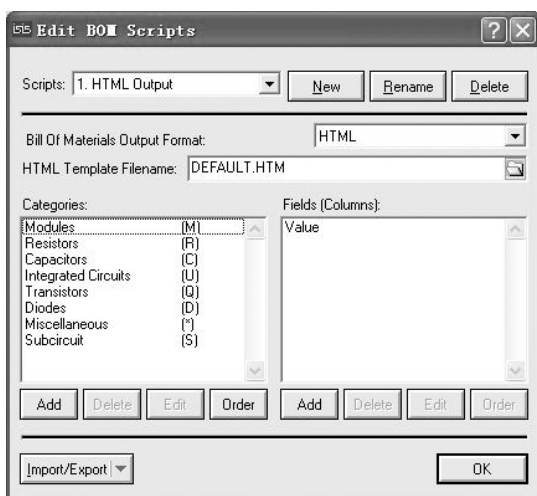


图 4-7 添加了“Subcircuit”项

在“Categories”列表框中选中“Subcircuit”，单击“Order”按钮，选择希望排序的对象，单击相应的按钮，即可实现排序，如图 4-8 所示。

同理，单击“Delete”、“Edit”等按钮，将出现对应的对话框，可对“Categories”及“Fields”进行添加、删除等操作。

2) 设置系统运行环境 在 PROTEUS ISIS 主界面中选择【System】→【Set Environment】菜单项，即可打开系统环境设置对话框，如图 4-9 所示。

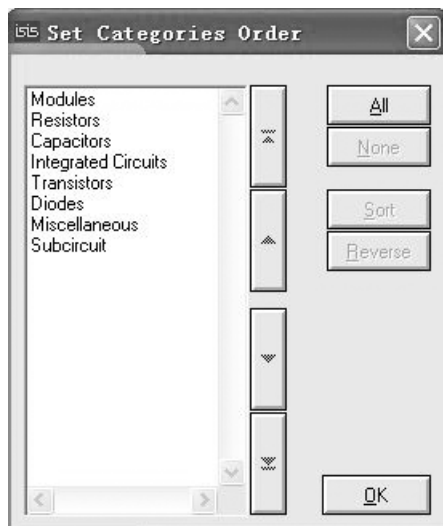


图 4-8 Order 对话框

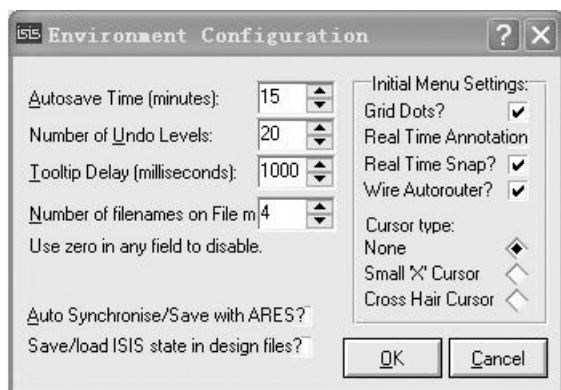


图 4-9 系统环境设置对话框

- Autosave Time (minutes): 系统自动保存时间设置 (单位为 min)。
- Number of Undo Levels: 可撤销操作的次数设置。
- Tooltip Delay (milliseconds): 工具提示延时 (单位为 ms)。
- Number of filenames on File Menu: File 菜单项中显示文件名的数量。
- Auto Synchronise/Save with ARES: 是否自动同步/保存 ARES。
- Save/load ISIS state in design files: 是否在设计文档中加载/保存 Proteus ISIS 状态。

3) 设置路径 选择【System】→【Set Paths】菜单项，即可打开路径设置对话框。如

图 4-10 所示为路径设置对话框，包括如下设置项。

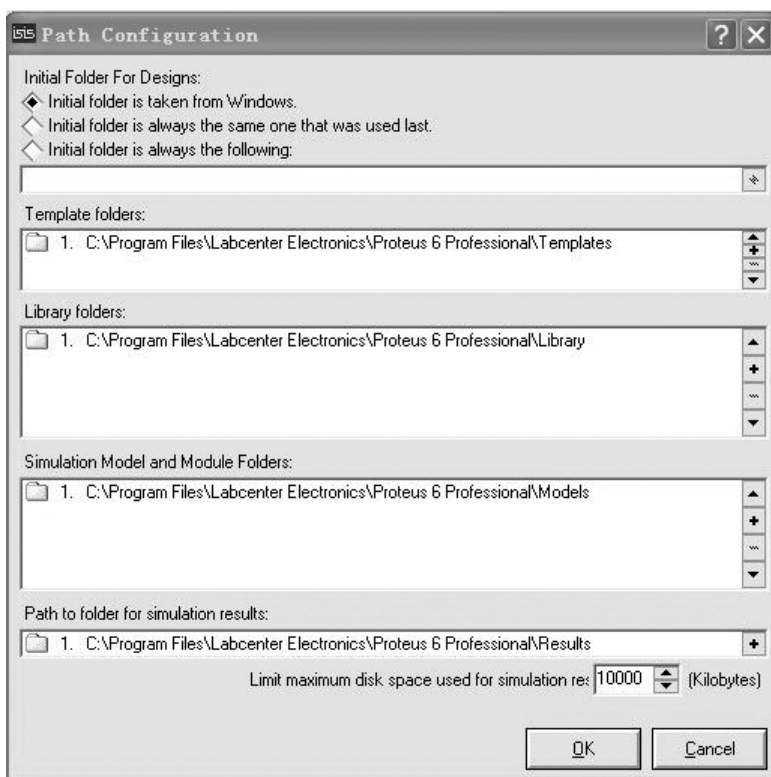


图 4-10 路径设置对话框

- Initial folder is taken from Windows: 表示从窗口中选择初始文件夹。
- Initial folder is always the same one that was used last: 表示初始文件夹为最后一次使用过的文件夹。
- Initial folder is always the following: 表示初始文件夹为下面文本框中输入的路径。
- Template folders: 表示模板文件夹路径。
- Library folders: 表示库文件夹路径。
- Simulation Model and Module Folders: 表示仿真模型及模块文件夹路径。
- Path to folder for simulation results: 表示仿真结果的存放文件夹路径。
- Limit maximum disk space used for simulation result (Kilobytes): 表示仿真结果占用的最大磁盘空间 (KB)。

4) 设置键盘快捷方式 选择【System】→【Set Keyboard Mapping】菜单项，打开键盘快捷方式设置对话框，如图 4-11 所示。

其中，在“Command Groups”下拉列表框中选择相应的选项，在“Available Commands”列表框中选择可用的命令，在该对话框下方的说明栏中显示所选命令的意义，“Key sequence for selected command”文本框中显示所选命令的快捷键。使用“Assign”和“Unassign”按钮可查看编辑或删除系统设置的快捷方式。单击“Options”旁的下三角按钮，选择“Reset to default map”选项即可恢复系统的默认设置，而选择“Export to file”选项可将上述键盘快捷方式导出到文件中，选择“Import from file”选项为从文件中导入图。

5) 设置 Animation 选项 选择【System】→【Set Animation Options】菜单项，即可打开仿真电路设置对话框，如图 4-12 所示。

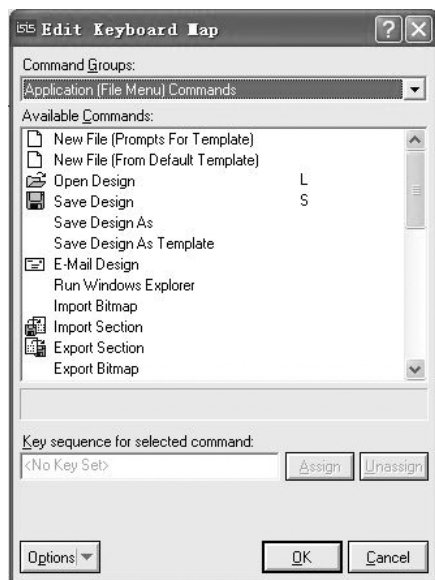


图 4-11 键盘快捷方式设置对话框

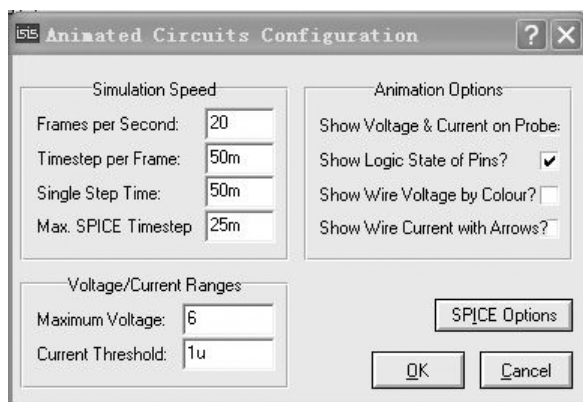


图 4-12 仿真电路设置对话框

- Show Voltage & Current on Probe: 是否在探测点显示电压值与电流值。
- Show Logic State of Pins: 是否显示引脚的逻辑状态。
- Show Wire Voltage by Colour: 是否用不同颜色表示导线的电压。
- Show Wire Current with Arrows: 是否用箭头表示导线的电流方向。

此外,单击“SPICE Options”按钮,弹出如图 4-13 所示对话框。

在该对话框中还可通过选择不同的选项来进一步对仿真电路进行设置。

**6) 设置仿真器选项** 选择【System】→【Set Simulator Options】菜单项,即可打开设置仿真器选项对话框,如图 4-14 所示。

在该对话框中可对仿真器选项进行设置,这里不再赘述。

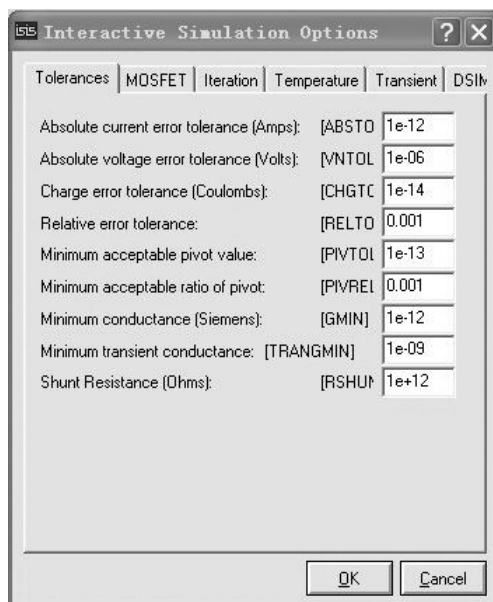


图 4-13 交互仿真选项对话框

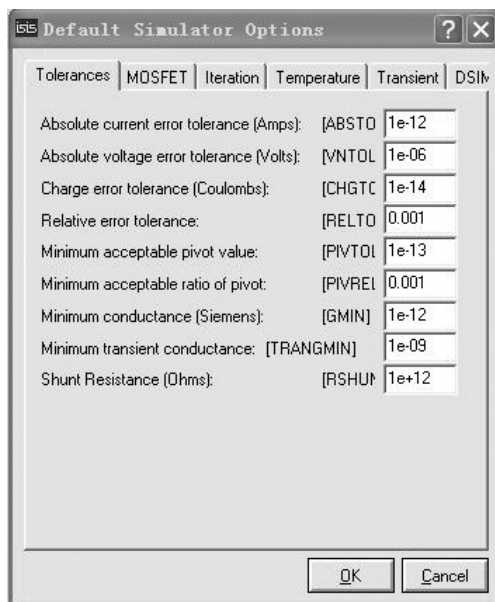


图 4-14 设置仿真器选项对话框



## 4.4 基于 PROTEUS 的电路设计

### 1. 设计流程

电路原理图设计流程如图 4-15 所示。

原理图设计步骤如下。

1) **新建设计文档** 在进入原理图设计之前,首先要构思好原理图,即必须知道所设计的项目需要由哪些电路来完成,用何种模板;然后在 PROTEUS ISIS 编辑环境中画出电路原理图。

2) **设置工作环境** 根据实际电路的复杂程度设置图纸的大小等。在电路图设计的整个过程中,图纸的大小可以不断地调整。设置合适的图纸大小是完成原理图设计的第一步。

3) **放置元器件** 首先从添加元器件对话框中选取需要添加的元器件,将其布置到图纸的合适位置,并对元器件的名称、标注进行设定;然后根据元器件之间的走线等联系对元器件在工作平面上的位置进行调整和修改,使原理图美观、易懂。

4) **对原理图进行布线** 根据实际电路的需要,利用 PROTEUS ISIS 编辑环境所提供的各种工具、命令进行布线,将工作平面上的元器件用导线连接起来,构成一幅完整的电路原理图。

5) **建立网络表** 在完成上述步骤之后,即可看到一张完整的电路图,但要完成印制板电路的设计,还需要生成一个网络表文件。网络表是印制板电路与电路原理图之间的纽带。

6) **原理图的电气规则检查** 当完成原理图布线后,利用 PROTEUS ISIS 编辑环境所提供的电气规则检查命令对设计进行检查,并根据系统提示的错误检查报告修改原理图。

7) **调整** 如果原理图已通过电气规则检查,那么原理图的设计就完成了,但是对于一般电路设计而言,尤其是较大的项目,通常需要对电路进行多次修改才能通过电气规则检查。

8) **存盘和输出报表** PROTEUS ISIS 提供了多种报表输出格式,同时可以对设计好的原理图和报表进行存盘和输出打印。

### 2. 设计实例

下面以简单电路为例,直观地介绍电路原理图的设计方法和步骤。

所需要的元器件如表 4-1 所示。

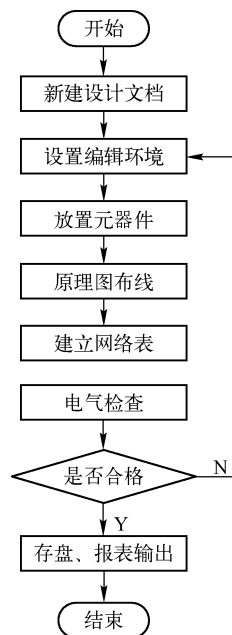


图 4-15 电路原理图设计流程

表 4-1 所需元器件

LAMP	电灯
BATTERY	电池组
SWITCH	开关
TES	电阻

所要设计的电路图如图 4-16 所示。

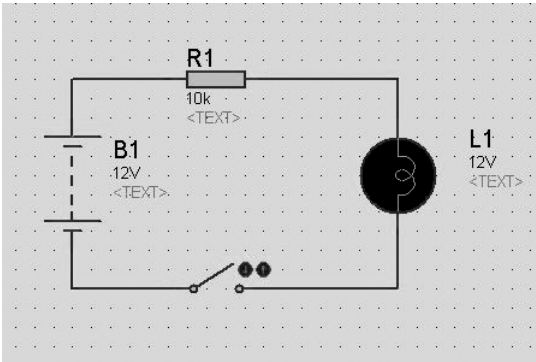


图 4-16 电路原理图

- (1) 创建一个新的设计文件。
- (2) 首先进入 PROTEUS ISIS 编辑环境。
- (3) 选择【File】→【New Design】菜单项，在弹出的模板对话框中选择“DEFAULT”模板，如图 4-17 所示，并将新建的设计保存在 E 盘根目录下，保存文件名为“example”。
- (4) 设置工作环境。
- (5) 打开【Template】菜单，对工作环境进行设置。
- (6) 在本例中，仅对图纸进行设置，其他项目使用系统默认的设置。
- (7) 选择【System】→【Set Sheet Sizes】菜单项，在出现的如图 4-18 所示的对话框中，选择“A4”复选框，单击“OK”按钮确认，即可完成页面设置。

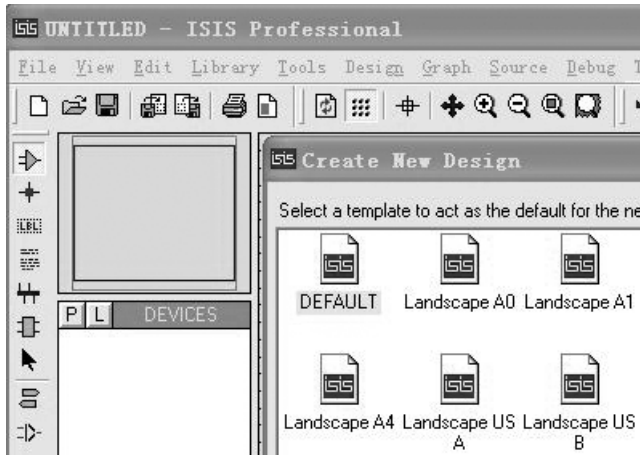


图 4-17 选择模板

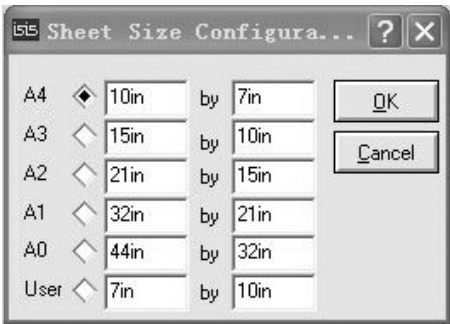


图 4-18 页面设置

(8) 拾取元器件，选择【Library】→【Pick Device/Symbol】菜单项，出现如图4-19所示对话框。

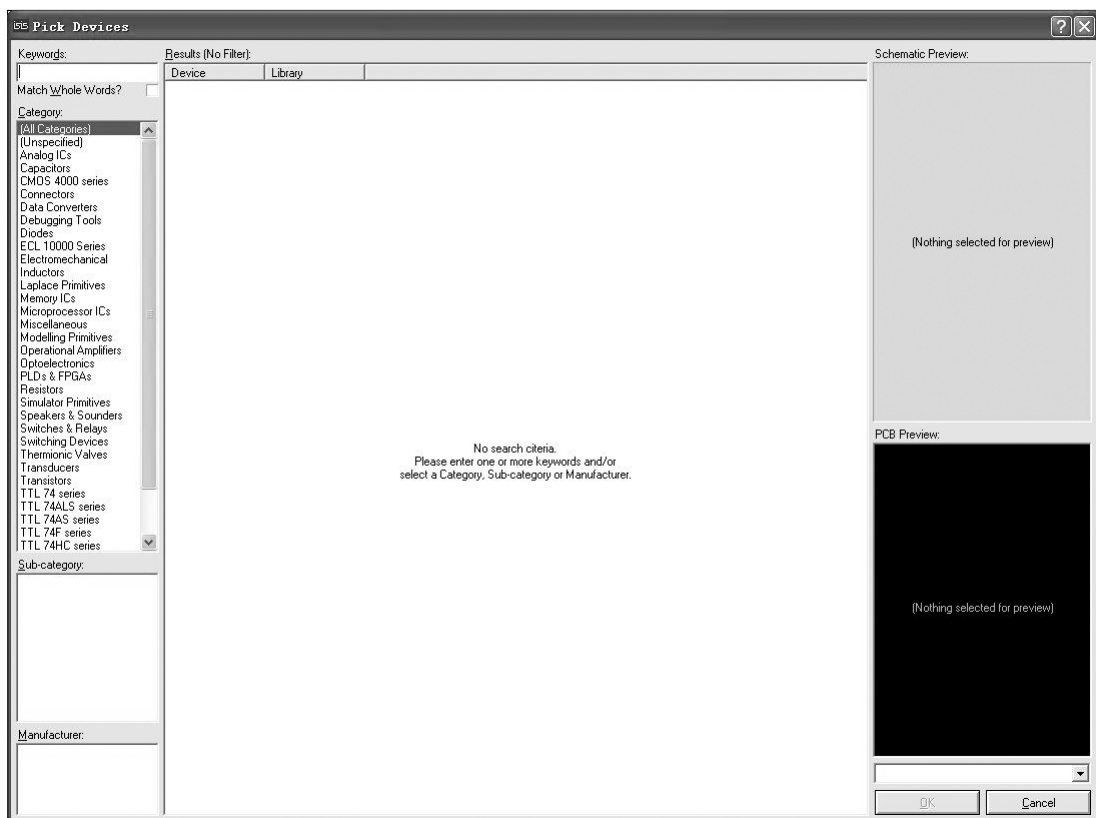


图4-19 元器件库

(9) 单击“OK”按钮，或在元器件列表区域双击元器件名称，即可完成对该元器件的添加。添加的元器件将出现在对象选择器列表中，如图4-20所示。

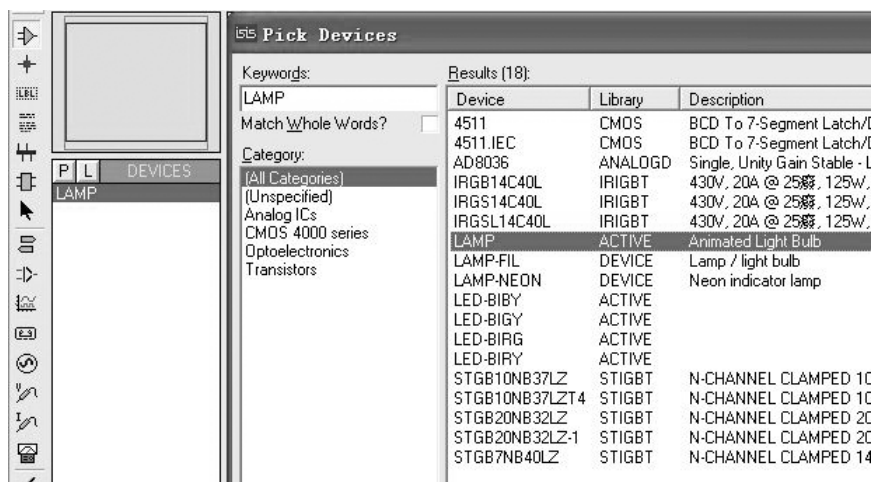


图4-20 元器件分类列表

(10) 在完成对元器件 LAMP 的查找后, 依次找到其他元器件。

(11) 在原理图中放置元器件。在当前设计文档的对象选择器中添加元器件后, 就要在原理图中放置元器件。下面以放置 LAMP 为例, 说明具体步骤。

① 选择对象选择器中的 LAMP 元器件, 在 PROTEUS ISIS 编辑环境主界面的预览窗口将出现 LAMP 的图标, 如图 4-21 所示。

② 在编辑窗口双击鼠标左键, 元器件 LAMP 被放置到原理图中。

③ 按照上述步骤, 分别将 BATTERY、SWITCH、RES 元器件放置到原理图中。

④ 将光标指向编辑窗口的元器件, 并单击该对象使其高亮显示。

⑤ 拖动该对象到合适的位置。

⑥ 调整好所有元器件后, 选择【View】→【Redraw】菜单项, 刷新屏幕, 此时图纸上有了全部元器件, 如图 4-22 所示。



图 4-21 LAMP 原理图表

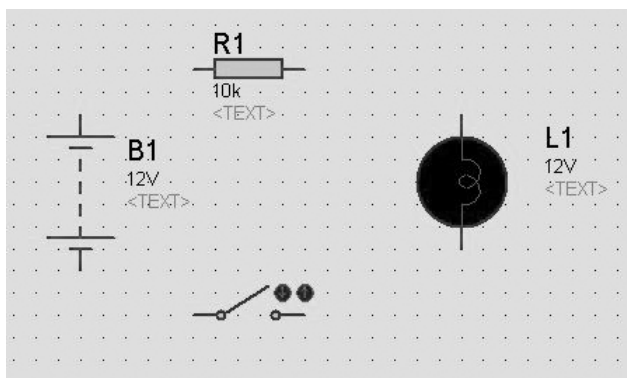


图 4-22 全部元器件

(12) 编辑元器件。放置好元器件后, 双击相应的元器件, 即可打开该元器件的编辑对话框。下面以 LAMP 的编辑对话框为例, 详细介绍元器件的编辑方式。LAMP 元器件的编辑步骤如下。

① 单击 LAMP 元器件, LAMP 将高亮显示。

② 再次单击 LAMP 元器件, 弹出如图 4-23 所示对话框, 编辑该元器件。

➤ Component Referer: 元器件在原理图中的参考号。

➤ Hidden: 选择元器件参考号是否出现在原理图中。

➤ Nominal Voltage: LAMP 的电压标称值。

➤ Resistance: LAMP 的阻抗。

③ 单击“OK”按钮, 结束元器件的编辑。

(13) 绘制原理图。PROTEUS ISIS 具有智能化特点, 在想要画线的时候能进行自动检测。在两个元器件间进行连线的步骤如下。

① 单击第一个对象连接点。

② 如果想让 PROTEUS ISIS 自动定出走线路径, 只需单击另一个连接点; 如果想自己决定走线路径, 只需在希望的拐点处单击。在此过程的任一阶段, 都可以按“Esc”键放弃画线。

按照上述步骤, 分别将 LAMP、BATTERY 及 POT - LIN 连线。连接后的原理图如图 4-24 所示。

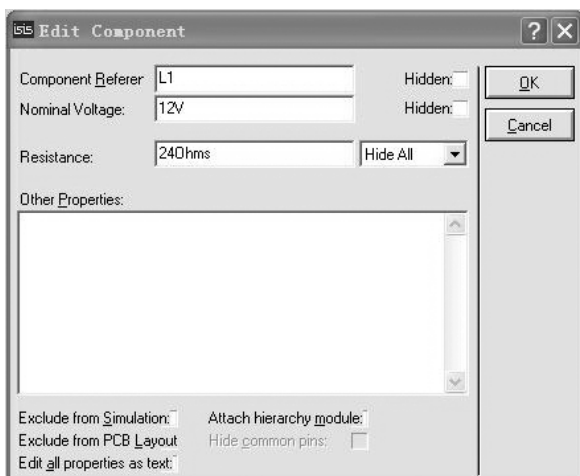


图 4-23 LAMP 元器件编辑对话框

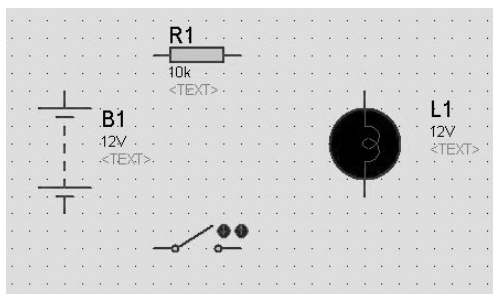


图 4-24 连接后的原理图

(14) 对原理图进行电气规则检查。选择【Tools】→【Electrical Rule Check】菜单项，出现电气规则检查报告单，如图 4-25 所示。

(15) 存盘及输出报表。将设计好的原理图文件存盘，同时，可使用【Tools】→【Bill of Materials】菜单项输出 BOM 文档，如图 4-26 所示。

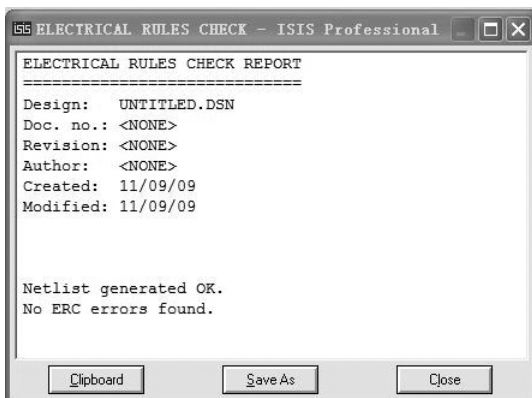


图 4-25 电气规则检查报告单

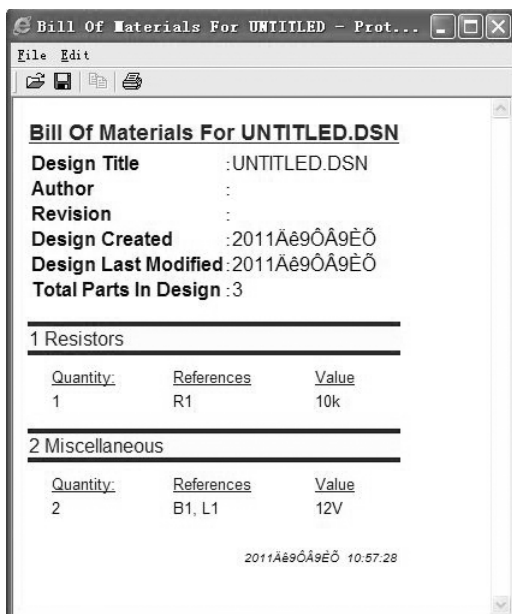


图 4-26 BOM 文档输出



## 4.5 元件创建

尽管 PROTEUS 提供了 8 000 多个元器件，但用户在使用过程中，还是要创建一些新器件，才能满足设计的要求，然而电路仿真模型的构建是一个复杂的过程，下面就原理图模



型、动态模型和 SPICE 模型的创建举例说明。对于 VSM 模型的构建，用户可以找 PROTEUS 总代理解决。

## 1. PROTEUS 模型分类

1) PROTEUS 的模型分为两大类——电气模型 (Electrical Model) 和绘图模型 (Graphical Model) 电气模型实现元件的电气特性，按规定的时序接收数据和输出数据；绘图模型实现仿真时与用户的交互，如 LCD/LED 显示、电动机等。一个元件可以只实现电气模型，也可以电气和绘图模型都实现。根据模型的创建方式，又可以分为原理图模型、SPICE 模型、VSM 模型和动态模型。

- 原理图模型 (Schematic Models): 由仿真原型构建，具有和被仿真器件等效电路行为的仿真模型 (并非器件的实际内电路)。PROTEUS ISIS 器件库中有大量的仿真原型，包括模拟和数字原型。
- SPICE 模型 (SPICE Models): 使用符合 SPICE3F5 的器件 SPICE 文件或库设计的仿真器件。SPICE 文件一般由器件制造商提供。
- VSM 模型 (VSM Models): 基于动态链接库 (DLL) 的仿真模型，DLL 是利用 Labcenter 提供的 VSM SDK (VSM 软件开发包) 用 C++ 编写的，用以描述器件电气行为。VSM 模型主要包括处理器等复杂逻辑器件和 LCM (液晶模块)、传感器等复杂外设器件。VSM SDK 需和 Labcenter 签订不扩散协议 (NDA) 后免费获得。
- 动态模型 (Active Components): 具有动画效果的动态模型，如继电器、灯、LED 数码显示等。通过动画模仿器件的动作过程。

2) 查看电路模型 以 LED-RED 为例，来说明如何查看电路的模型。打开 PROTEUS 编辑界面，单击 P 按钮，在出现的对话框中输入 “LED-RED”，即可看到如图 4-27 所示的元器件属性窗口。

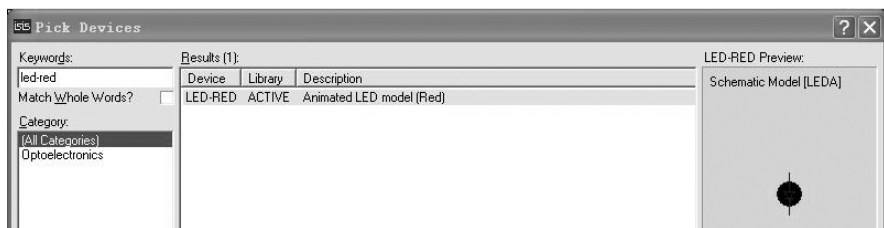


图 4-27 在元器件窗口中查看模型类型

## 2. 原理图模型创建

原理图模型创建过程分为三个步骤，如图 4-28 所示。

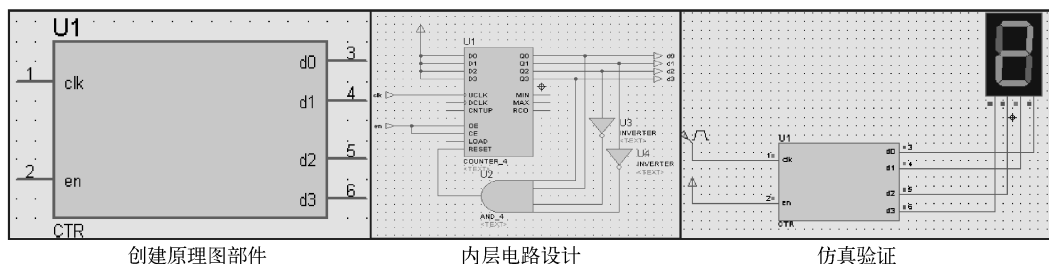


图 4-28 原理图模型创建过程

下面以二选一为例说明如何创建原理图模型。

### 1) 创建原理图部件

(1) 绘制原理图符号。单击绘图工具按钮 (2D graphics box)，在 ISIS 编辑区画一个适当大小的方框，在创建的原理图部件中右击，在弹出的菜单中选择“PLACE”→“Device Pin”→“Default”，并将引脚放置到框的适当位置（带×的一端放在外），图 4-29 是创建的二选一原理图部件。

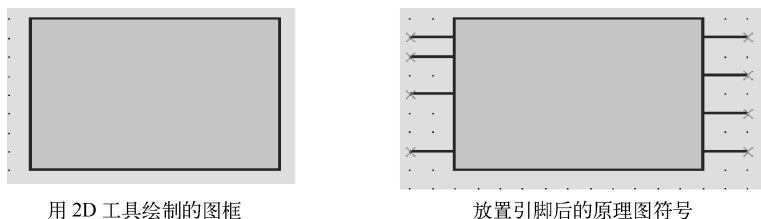


图 4-29 创建原理图部件

(2) 编辑引脚。右击原理图符号的引脚再左击，弹出如图 4-30 所示的编辑引脚对话框，对每个引脚按表 4-2 所示进行编辑。

表 4-2 编辑引脚

编 号	名 称	显示引脚	显示名称	显示编号	引脚电气类型
1	A	√	√	√	IP
2	B	√	√	√	IP
3	S	√	√	√	IP
4	GND	×	×	×	PP
5	NC	×	×	×	PS
6	NC	×	×	×	PS
7	Y0	√	√	√	OP
8	VDD	×	×	×	PP

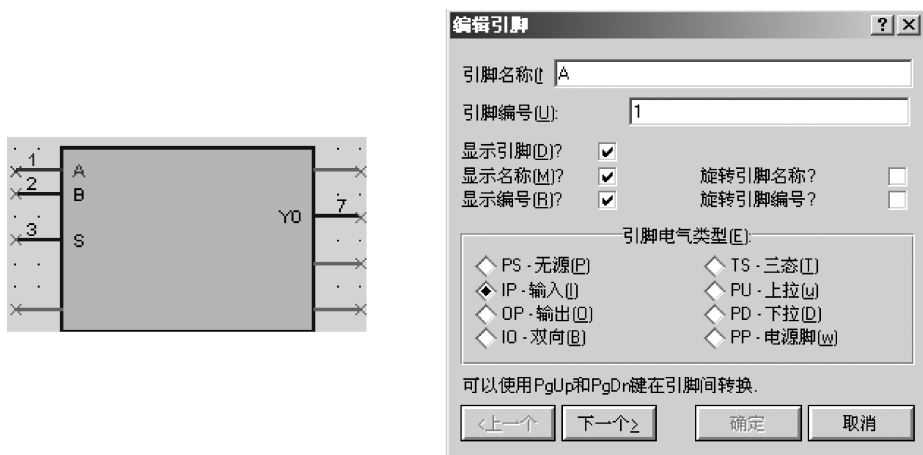


图 4-30 编辑引脚后的原理图符号

对于原理图中的引脚编辑应注意：

- 每个引脚必须有名称；
- 相同名称的引脚，系统认为是电气相连的；
- 对 $\overline{\text{WR}}$ 类引脚名，在引脚名前、后加符号\$即可，如\$WR\$，就能出现带非的符号。

(3) 定义器件名称。选中原理图符号，单击“库”→“制作器件”命令，弹出如图 4-31 所示的器件属性对话框，定义本设计的名称为“MUX21A”，参考前缀为“MU”（为了查找定义一个前缀）。

(4) 指定器件封装。在制作器件对话框中单击“下一个”按钮，弹出 Packagings 界面，进入元器件封装设置。单击“添加/删除”按钮，再单击“添加”按钮，进入 Pick Packages 对话框，选择封装为“DIL08”，结果如图 4-32 所示。单击“指定封装”按钮，再次进入 Packagings 界面确定封装，结果如图 4-33 所示。单击“下一步”按钮，弹出 Component Properties & Definitions 界面进入下一步骤。



图 4-31 定义设计名称

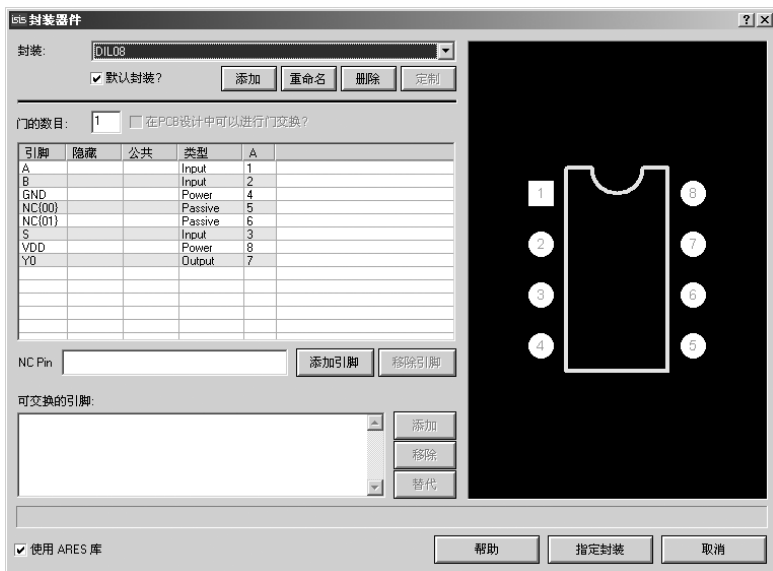


图 4-32 指定封装对话框

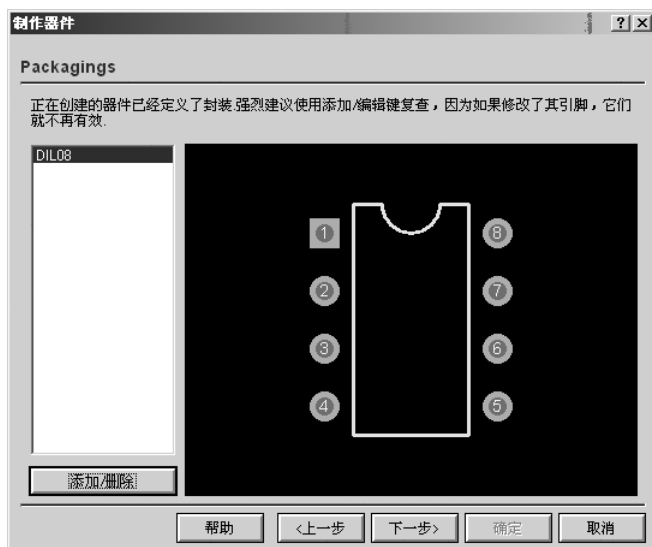


图 4-33 确定封装

(5) 定义元器件属性。在如图 4-34 所示的元器件属性及定义对话框中，使用“新建”和“删除”按钮编辑元器件属性，这些属性用来指定 PCB 封装、仿真模型参数及一些其他信息等。

(6) 指定元器件存放库。在图 4-34 所示对话框中单击“下一步”按钮，再单击“下一步”按钮，出现如图 4-35 所示对话框，右边是保存元器件库选择，在左边的“器件类”栏单击“新建”按钮，在弹出的对话框中填写“JSMLIB”，然后单击“确定”按钮。至此，器件制作完成。



图 4-34 元器件属性及定义对话框



图 4-35 定义元器件存放库对话框

## 2) 内层电路设计

(1) 转入内部电路。从 ISIS 窗口的对象列表框中选择器件 MUX21A 放到 ISIS 编辑区，打开器件的属性对话框，选中附加层次模块，单击“确定”按钮退出属性对话框。然后单击“Design”选择“Goto Sheet”转入子页面。

(2) 内层电路设计。在子页面中绘制如图 4-36 所示的电路原理图，值得注意的是：子页中的引脚端口要与父电路的引脚端口一样，还有电源应加进去。

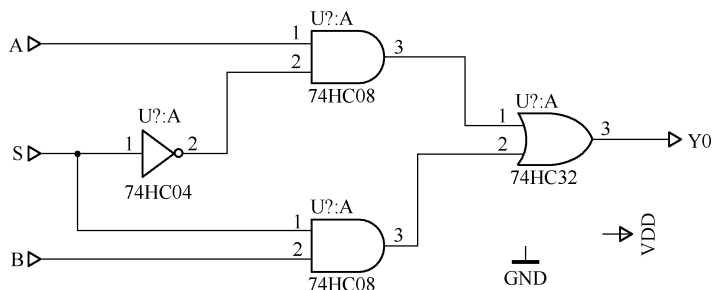


图 4-36 内部电路原理图

## 3) 验证电路功能

(1) 验证电路功能。完成子电路后再返回父电路中，在父电路中对电路的功能进行验证。如图 4-37 所示，设在 A 端和 B 端分别加上两种不同频率的方波，S 端接选择开关，用示波器观察输出点的波形（A 端接 100Hz 方波，B 端接 1 000Hz 方波，S 端接开关）。

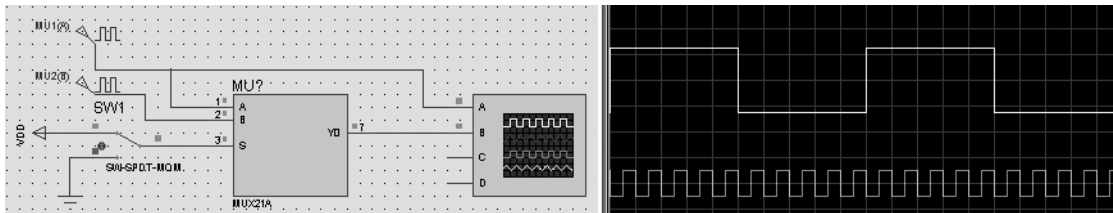


图 4-37 验证电路及测试波形

(2) 生成和加载模板文件。返回内部电路，在内部电路设计页单击“工具”→“编译模型”，在弹出的对话框中保存模型文件为 D:\ \ JSM \ \ MUX21A. MDF。

(3) 返回到父页，选中 MUX21A，单击右键，选择创建器件，然后单击“下一步”按钮，直到出现如图 4-38 所示界面。按图 4-38 所示填好内容，单击“下一步”按钮，直到完成。至此，一个元器件的模型就创建成功了。



图 4-38 二选一电路模型文件加载

下面再通过 DAC0832 的创建来加深对原理图模块创建的认识。

DAC0832 是双列直插式 8 位 D/A 转换器，其内部结构和引脚分布如图 4-39 所示。

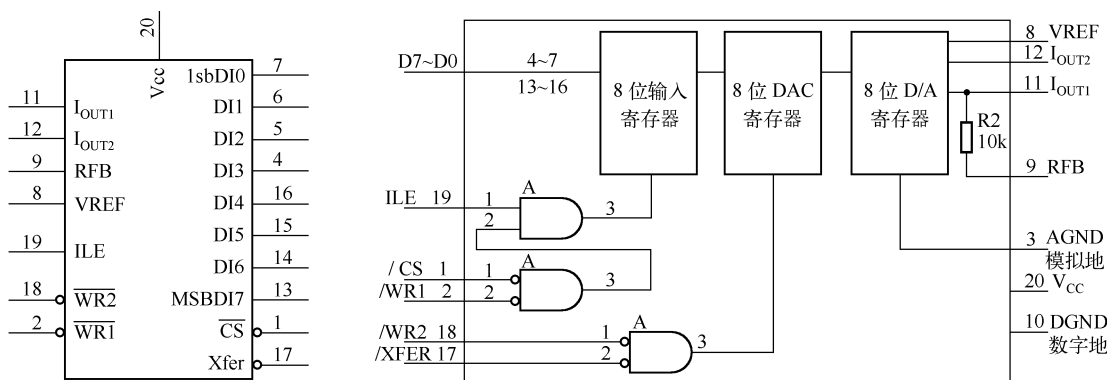


图 4-39 DAC0832 内部结构和引脚分布

- DI7 ~ DI0: 8 位的数据输入端，DI7 为最高位。
- I<sub>OUT1</sub>、I<sub>OUT2</sub>: 模拟电流输出端，当 DAC 寄存器中的数据全为 1 时，输出电流最大；当 DAC 寄存器中的数据全为 0 时，输出电流为 0。I<sub>OUT1</sub> + I<sub>OUT2</sub> = 常数。
- RFB: 反馈电阻引出端，DAC0832 内部已经有反馈电阻，所以 RFB 端可以直接接到外部运算放大器的输出端，这样相当于将一个反馈电阻接在运算放大器的输出端和输入端之间。
- VREF: 参考电压输入端，此端可接一个正电压，也可接一个负电压，它决定 0 ~ 255 的数字量转化出来的模拟量电压值的幅度，VREF 的范围为 -10 ~ +10V。

➤  $V_{cc}$ 、AGND：芯片供电电压，范围为 5 ~ 15V，AGND 是模拟量地。

➤ DGND：数字量地。

1) 搭建元件的测试平台 先设计 DAC0832 的原理图符号，生成一个不含模型的器件，并编辑其属性。勾选“附件层次化模块”，这样，该器件就可以进入内层进行设计。按照数据手册的说明，搭建一个简单的测试平台，如图 4-40 所示。可以想象，图中 DAC0832 将输出一个锯齿波。

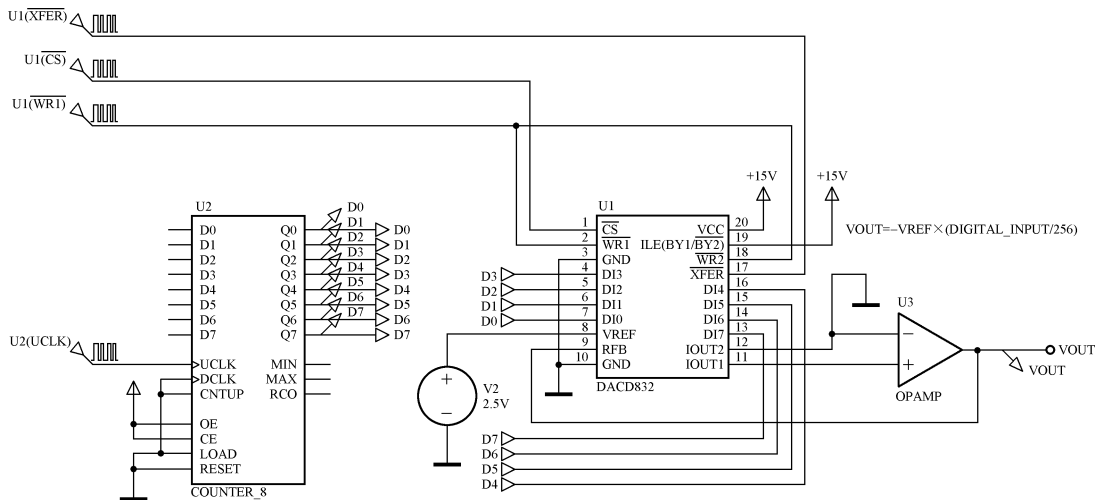


图 4-40 DAC0832 仿真测试电路

2) 层次化设计内部电路 进入 DAC0832 的内层（单击右键，选择“转到子页面”命令），由于 DAC0832 的功能比较复杂，设计中对 T 形网络（LADDER）部分采用模块电路。在器件库的 Modeling Primitives 中，有很多的基本逻辑电路原型和模拟器件原型，下面按照 DAC0832 的逻辑功能构建其等效电路，主要包括两级锁存器和 T 形网络。图 4-41 为 DAC0832 的内层电路，图 4-42 为 T 形网络模块电路的内部结构。

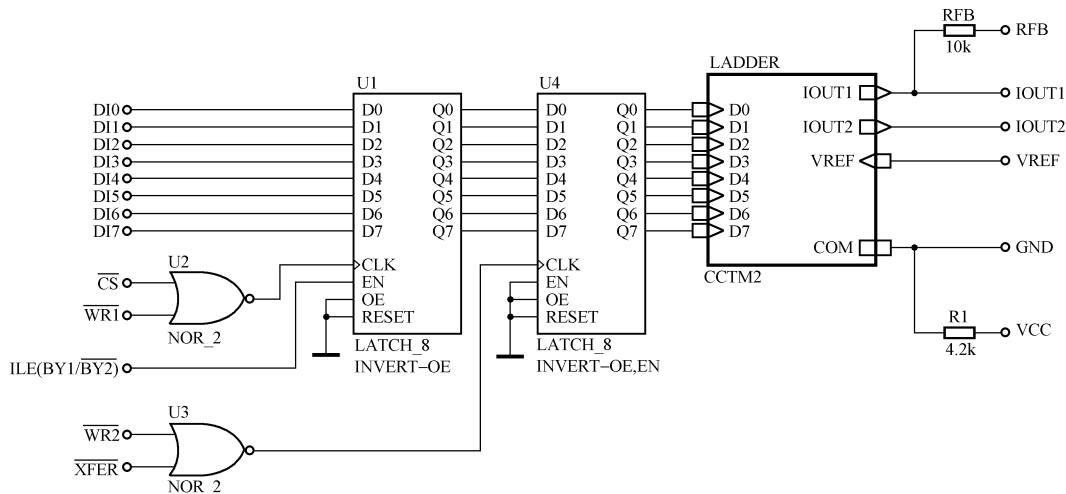


图 4-41 DAC0832 的内层电路

T 形网络构成了 DA 转换的核心部分，梯形网络完全参考数据手册来设计，主要采用数控开关（DSWITCH）和参数化电阻原型，电阻值在图中的脚本文件中定义，格式为：

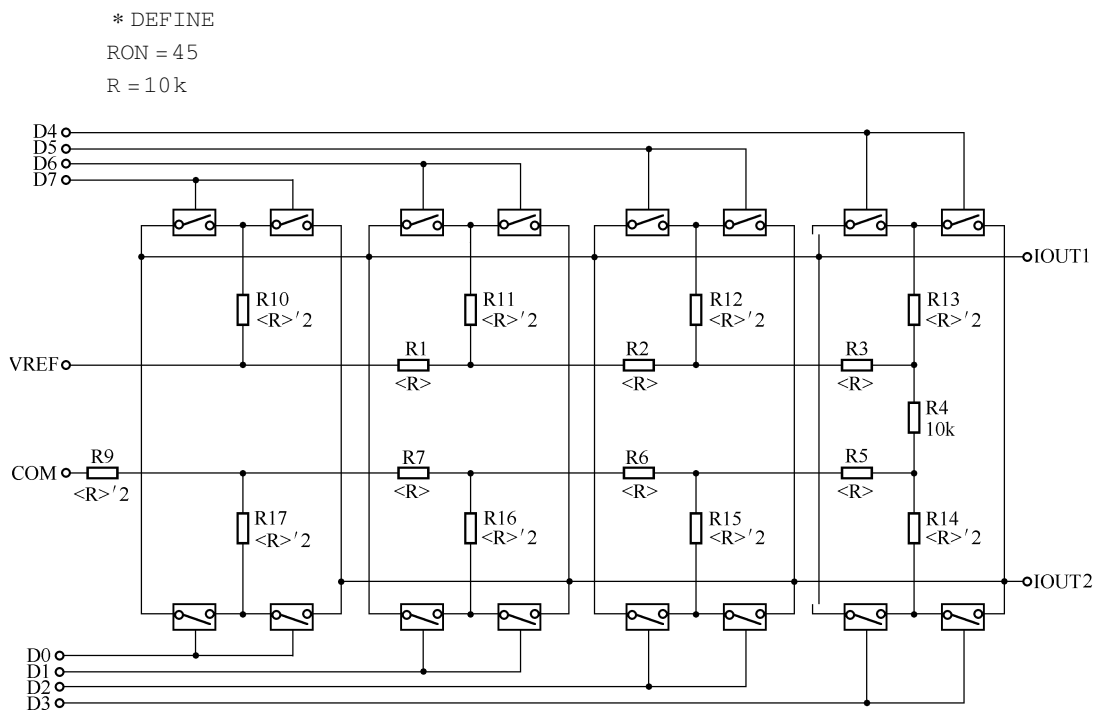


图 4-42 T 形网络模块电路的内部结构

**3) 模型元件功能测试** 为了清晰地观察逻辑时序, 我们采用混合图表来测量电路的时序和波形, 如图 4-43 所示。

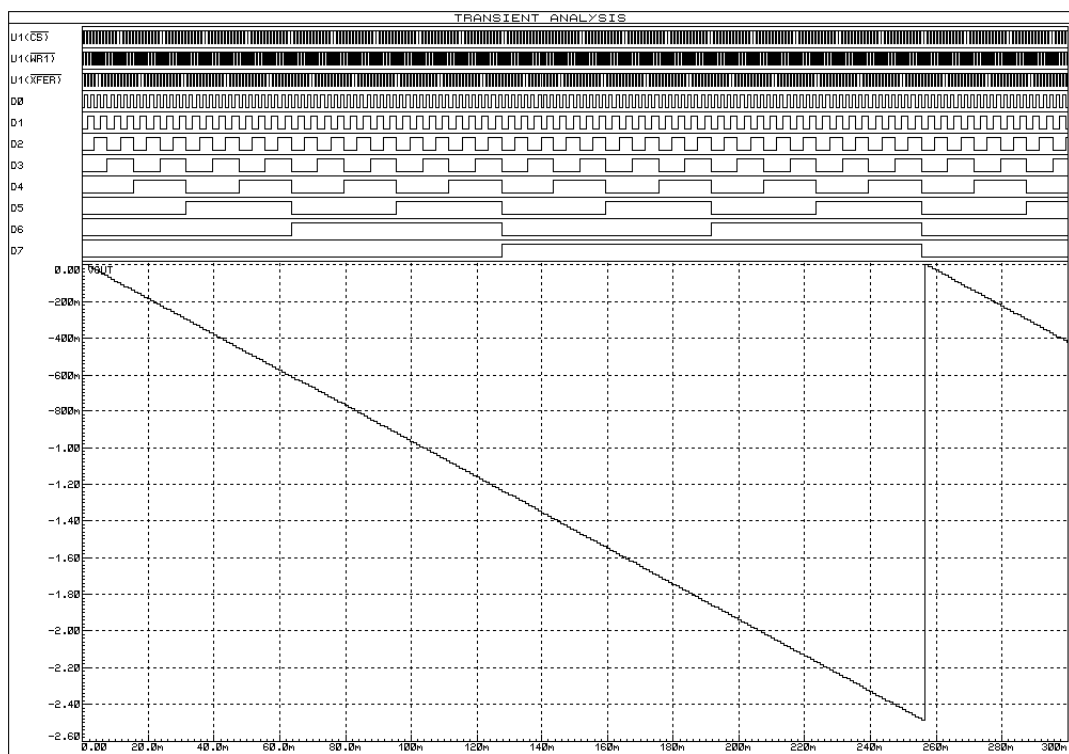


图 4-43 功能测试结果



4) **编译生成模型文件** 功能测试完成后, 接下来就可以对 DAC0832 的内电路进行编译了。进入 DAC0832 的内层电路, 选择“工具”菜单下的“模型编译器”命令, 生成一个 DAC0832.mdf 文件 (模型描述文件)。然后选中 DAC0832, 在右键菜单中选择“创建器件”命令, 在“Component Properties & Definitions”页面中添加“MODFILE”属性, 取默认值为“DAC0832”, 如图 4-44 所示。



图 4-44 创建仿真器件

5) **导入库** 把生成的模型封装入库, 这样就完成了 DAC0832 仿真模型的设计。

### 3. SPICE 模型创建

1) **SPICE 文件** SPICE 文件是一个描述器件性能指标的文本文件, 一般由器件制造商提供。PROTEUS 使用兼容工业标准 SPICE3F5 的文件, 下面是厂家提供的电压基准芯片 KA431 的 SPICE 文件。

```
* REFERENCE
*
* | ANODE
* | | CATHODE
* | | |
.SUBCKT KA431 1 2 3
V1 6 7 DC 1.4V
I1 2 4 1E-3
R1 1 2 1.2E6
R2 4 2 RMOD 2.495E3
R3 5 7 .2
D1 3 6 DMOD1
D2 2 3 DMOD1
D3 2 7 DMOD2
E1 5 2 POLY(2) (4,2) (1,2) 0 710 -710
.MODEL RMOD RES (TC1 = 1.4E-5 TC2 = -1E-6)
.MODEL DMOD1 D (RS = .3)
.MODEL DMOD2 D (RS = 1E-6)
.ENDS
```

2) SPICE 模型创建过程 SPICE 模型创建过程如图 4-45 所示。



图 4-45 SPICE 模型创建过程

(1) 设计器件的原理图符号。根据 SPICE 文件的引脚名称，用原理图模型创建的方法，得到如图 4-46 所示的原理图符号。

(2) 加入 SPICE 文件设定。在编辑窗口右击，弹出如图 4-47 所示的窗口，然后编辑元件属性。

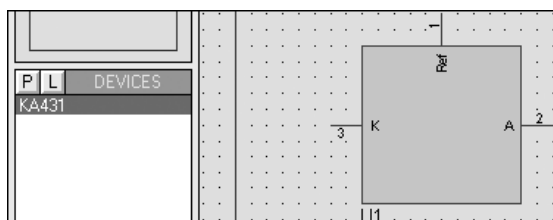


图 4-46 KA431 原理图符号

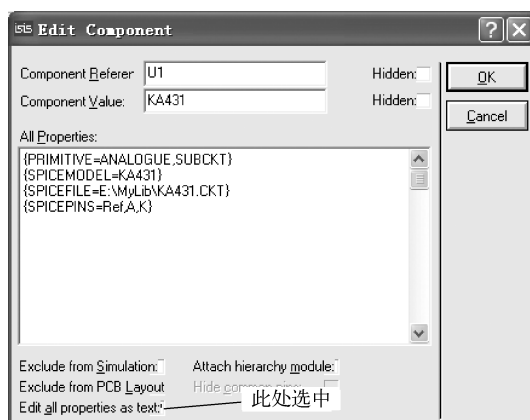


图 4-47 编辑元件属性

(3) 仿真测试。如图 4-48 所示，对元器件进行功能测试。

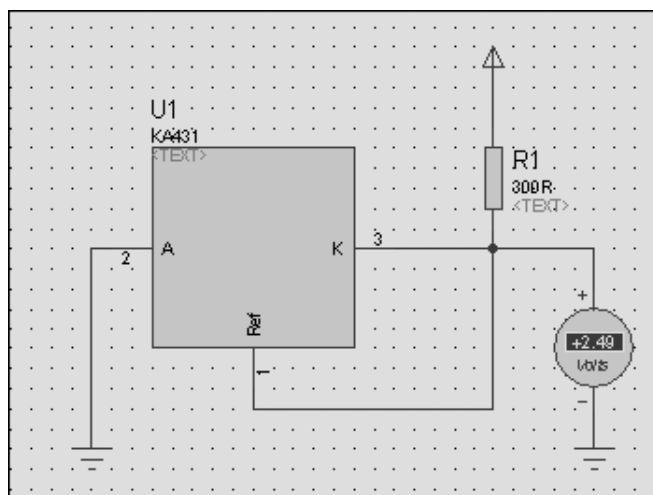


图 4-48 测试元件功能

#### 4. VSM 模型创建

对于 MCU 等较为复杂的仿真模型，PROTEUS 提供了一些 C++ 抽象类接口，可以利用 Labcenter 提供的 VSM SDK 用 C++ 开发，基本过程如下。

- (1) 用 2D 工具绘制器件的原理图符号及图形、引脚。
- (2) 创建器件并设置动态器件 (Active Component) 的属性, 选择使用 DLL。
- (3) 在 C++ 环境下引用 SDK 提供的头文件, 参考器件的 datasheet 设计器件的动态链接库 (DLL), 实现器件的电气和图形行为。
- (4) 将 DLL 文件设置到器件的属性中, 进行仿真测试。

## 5. 动态模型创建

动态仿真模型在 PROTEUS 中作为与用户交互的一种工具, 在仿真的过程中显得尤其重要。无论是它的外形, 还是对外部事件的响应, 对整个仿真的效果都有着深远的影响, 下面以 7 段数码管和开关为例讲述动态模型的建立过程。

1) 7 段数码管的建立 用 2D 工具绘制各笔段, 得到如图 4-49 所示符号图形。

图 4-49 所示的数码管有 7 个组成元素, 对于每个元素而言, 它有高电平和低电平两种状态, 所有的元素组成在一起就能表示数码管的总状态。在 7 段数码管模型中, 我们按图 4-50 所示动态模型对话框中的设置来定义数码管的位。

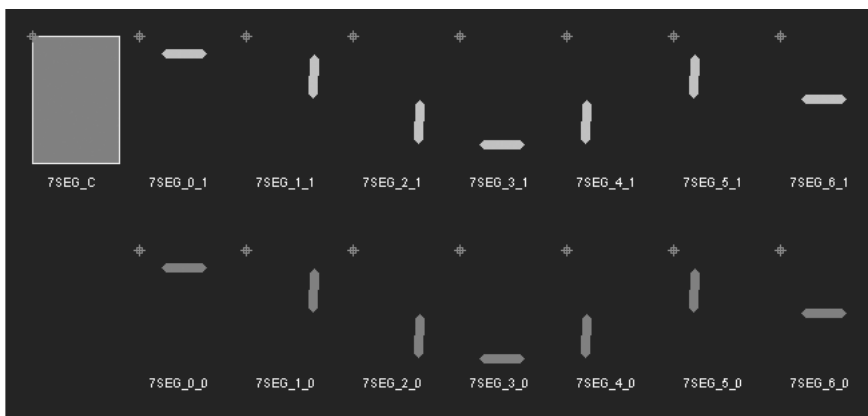


图 4-49 符号图形

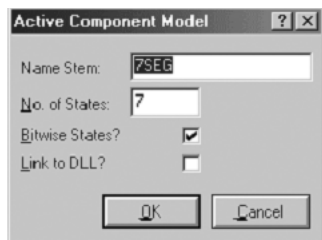


图 4-50 属性设置

为了更清楚地展现出做模型的全过程, 我们将整个 7SEG 元件分解, 得到以上的情况。以数码管的第 0 位来说, 当这个引脚上为低电平时显示 7SEG0\_0 的这个状态, 为高电平时显示 7SEG0\_1 的这个状态。依次画好以后, 为了得到一个完整的仿真模型, 我们采用实时探针 PRIMITIVE = DIGITAL, RTVPROBE。

然后创建数码管的 7 个引脚, 分别命名为 D0 ~ D6, 这样就足够了。然而, 要使元件具有模拟 LED 器件的特性, 我们有必要在此基础上建立一个原理图模型, 如图 4-51 所示。

这样每个引脚的电流都会通过独立的实时电流探针来测量, 而组成数码管每个元素的探针参数则被用来决定哪一段图形将被控制。

2) 开关模型的建立 开关在一般的仿真中也是用得比较多的一个交互式仿真器件。

开关的制作是相对比较简单, 它由仿真原型 RTSWITCH 建模得到, 所以不需要什么原理图模型。同样, 这个开关器件也有两种状态, 即由两种状态元素组成。

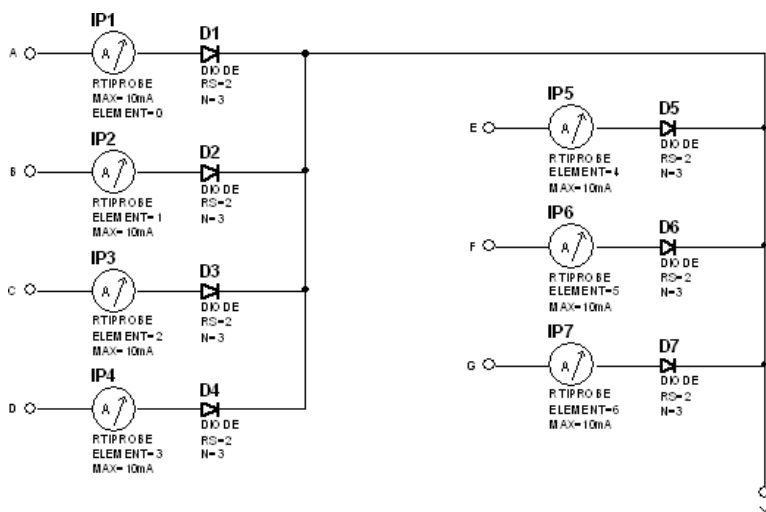


图 4-51 数码管内部模拟电路

首先在原理图中绘制出两个元件 SWITCH\_0 和 SWITCH\_1，再制作整个模型，包括引脚和 TOGGLE，如图 4-52 所示。



图 4-52 开关模型符号图形

SWITCH 绘制好以后，开始制作器件，设置其参数，如表 4-3 所示。

表 4-3 开关模型参数设置

名 称	描 述	数据 类型	编辑 模式	默 认 值
R(0)	Off Resistance	FLOAT (PNZ)	NORMAL	100MΩ
R(1)	On Resistance	FLOAT (PNZ)	NORMAL	0.1Ω
TSWITCH	Switching Time	FLOAT (PNZ)	NORMAL	1ms
PRIMITIVE	Primitive Type	STRING	HIDDEN	PASSIVE, RTSWITCH
STATE	Active State	INTEGER	HIDDEN	0

参数设置完毕后，就可以按图 4-53 所示电路对开关进行测试了。  
电路测试正确后，就完成了开关模型的制作。

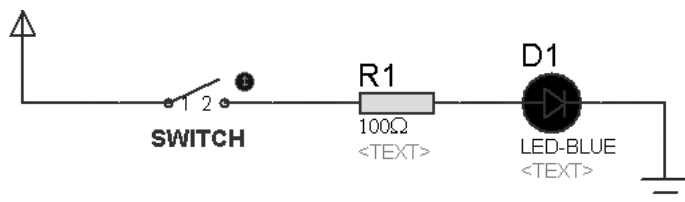



图 4-53 开关模型测试电路




## 4.6 PROTEUS ISIS 原理图输入

### 1. 元件查找

从元器件库中选择元器件。

(1) 从工具箱中选择“Component”图标.

(2) 单击对象选择器中的“P”按钮，弹出“Pick Devices”(元件拾取)对话框



(3) 在“Keywords”文本框中输入一个或多个关键字，或使用元器件类列表和元器件子类列表，过滤掉不希望出现的元器件，同时定位希望出现的元器件。

(4) 在元件列表区域双击元器件，即可将元器件添加到设计中。

(5) 当完成元器件的提取后，单击“OK”按钮关闭对话框，并返回 PROTEUS ISIS。

### 2. 元件放置




(1) 从工具箱中选择元器件图标。如果用户需要的元器件在对象选择器中未列出，则必须从元器件库中提取。

(2) 在对象选择器中选中需要的元器件。在 PROTEUS ISIS 的预览窗口可预览所选中的元器件。


(3) 在编辑窗口中希望元器件出现的位置双击，即可放置元器件。还可先单击然后对其进行拖动操作。

(4) 根据需要，使用旋转及镜像按钮确定元器件的方位。

### 3. 连线

**1) 基本电路连线** 电路连线采用按格点捕捉和自动连线的形式，所以首先应确定编辑窗口上方的自动连线图标和自动捕捉图标为按下状态。PROTEUS 的连线是非常智能的，它会判断下一步的操作是否想连线从而自动连线，而不需要选择连线的操作。用鼠标左键单击编辑区元件的一个端点拖动到要连接的另外一个元件的端点，先松开左键后再单击鼠标左键，即完成一根连线。如果要删除一根连线，右键双击连线即可。连线完成后，如果再想回到拾取元件状态，可按下左侧工具栏中的“元件拾取”图标.

#### 2) 总线模式：放置总线

(1) 从工具箱中选择总线“Bus”图标.

(2) 在希望总线起始端出现的位置单击鼠标左键。

(3) 在希望总线路径的拐点处单击鼠标左键。

(4) 在总线的终点单击鼠标左键，然后单击鼠标右键，可结束总线放置。

### 4. 标注

**1) 文本编辑** PROTEUS ISIS 的一个重要的特色就是支持自由格式的文本编辑 (Text Script)，其使用方法如下：

- 定义变量，用于表达式或作为参数；
- 定义原始模型及脚本，用于 VSM 仿真；
- 标注设计；
- 当某一元器件被分解时，用于保存和封装信息。


#### (1) 放置和编辑脚本。

- ① 从工具箱中选择 Text Script 图标，在编辑窗口单击，弹出“Edit Script Block”对话框。
- ② 在该对话框中选择“Script”选项卡，如图 4-54 所示。
- ③ 在“Text”区域输入文本，同时，选择“Style”选项卡，还可以在此选项卡中调整“Script”的属性。
- ④ 单击“OK”按钮，完成操作。

#### (2) 编辑脚本。

- ① 左击要编辑的脚本，选中该脚本，然后单击该脚本打开“Edit Script Block”对话框；或将光标放置在要编辑的脚本上，使用组合键“Ctrl + E”打开“Edit Script Block”对话框。
- ② 根据需要调整脚本属性。可对“Edit Script Block”对话框包含的两个选项卡“Script”和“Style”分别进行编辑。
- ③ 编辑完成后，单击“OK”按钮或按组合键“Ctrl + Enter”保存修改。

### 2) 连线标签模式

- (1) 从工具箱中选择“Wire Label”图标。

- (2) 将鼠标指针指向希望放置标签的总线分支位置，被选中的导线变成虚线，鼠标指针处出现一个“×”，此时单击鼠标左键，打开“Edit Wire Label”对话框，如图 4-55 所示。

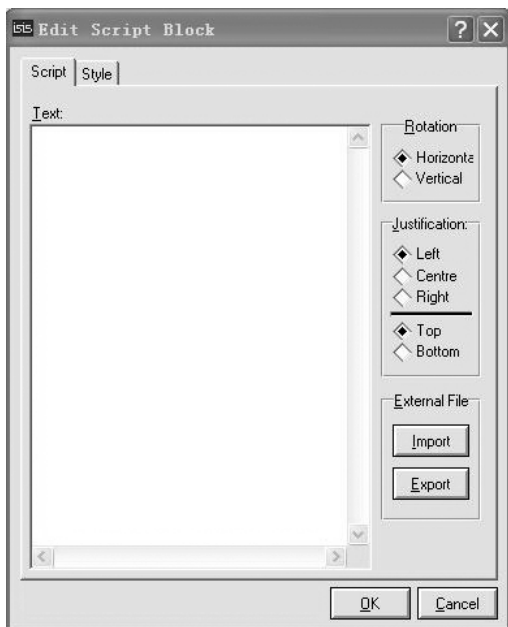


图 4-54 “Script”编辑选项卡

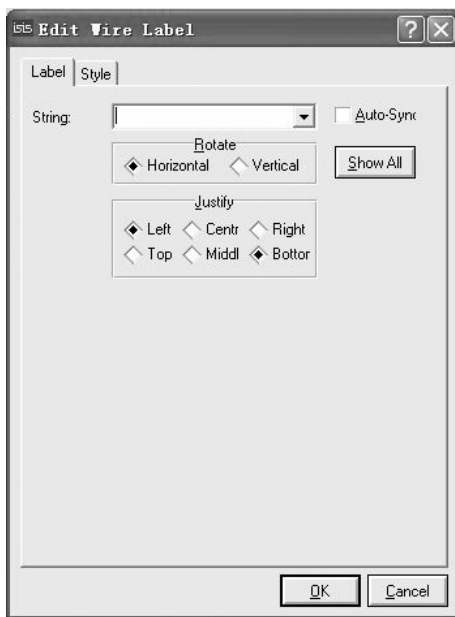


图 4-55 连线标签编辑对话框

- (3) 在该对话框的“Label”选项卡中输入相应的文本，如“AD0”。
- (4) 单击“OK”按钮，结束文本的输入。

在放置相邻的第二个总线标签时，系统不会像 Protel 软件那样自动按序标出文本号，而

读者只需连续单击“OK”按钮即可。实际情况是，我们必须重新输入一次文本，或单击“Edit Wire Label”对话框中“String”右侧的下拉按钮，当出现“AD0”时，将其修改成“AD1”，相对省事些，如图 4-56 所示。

像删除元件一样直接双击右键来删除标签是不行的，那样会使它所连接的导线一起被删除。想更改或删除总线标签可以对准总线标签单击右键，打开其快捷菜单，如图 4-57 所示。其中第一项“Edit Label”是编辑总线标签，第二项“Delete Label”是删除总线标签，第三项“Drag Wire”是移动总线标签。

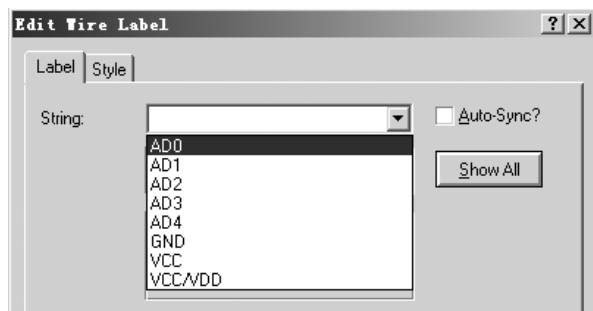


图 4-56 连线标签编辑下拉列表

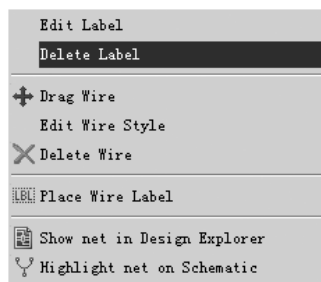


图 4-57 删除连线标签菜单命令










## 4.7 虚拟仿真工具

### 1. 激励源

PROTEUS ISIS 为用户提供了 14 种激励源，如表 4-4 所示。

表 4-4 激励源

名 称	符 号	意 义
DC	<TEXT> 	直流信号发生器
SINE	<TEXT> 	正弦波信号发生器
PULSE	<TEXT> 	脉冲发生器
EXP	<TEXT> 	指数脉冲发生器
SFFM	<TEXT> 	单频率调频波发生器
PWLIN	<TEXT> 	分段线性激励源
FILE	<TEXT> 	FILE 信号发生器

续表

名 称	符 号	意 义
AUDIO		音频信号发生器
DSTATE		数字单稳态逻辑电平发生器
DEDGE		数字单边沿信号发生器
DPULSE		单周期数字脉冲发生器
DCLOCK		数字时钟信号发生器
DPATTERN		数字模拟信号发生器
SCRIPTABLE		可编辑信号发生器

用户可根据需要对其进行属性设置。下面以直流信号和正弦波信号为例，对其属性进行设置。

1) 直流信号发生器 直流信号发生器用来产生模拟直流电压或电流。

(1) 放置直流信号发生器。

① 在 PROTEUS ISIS 环境下单击工具箱中的“Generator Mode”按钮图标，出现如图 4-58 所示的所有激励源的名称列表。

② 用鼠标左键单击“DC”，则在预览窗口出现直流信号发生器的符号，如图 4-59 所示。



图 4-58 激励源名称列表

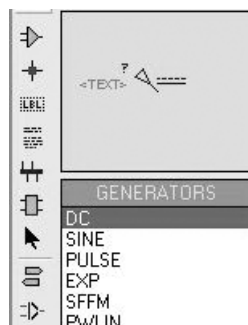


图 4-59 激励符号

③ 在编辑窗口双击，则直流信号发生器被放置到原理图编辑界面中。可使用镜像、翻转工具调整直流信号发生器在原理图中的位置。

(2) 直流信号发生器属性设置。

① 在原理图编辑区中，用鼠标左键双击直流信号发生器符号，出现如图 4-60 所示的属性设置对话框。

② 默认为直流电压源，可以在右侧设置电压源的大小。

(3) 如果需要直流电流源，则在图 4-60 中选中左侧下面的“Current Source”，右侧自动出现电流值的标记，根据需要填写即可，如图 4-61 所示。



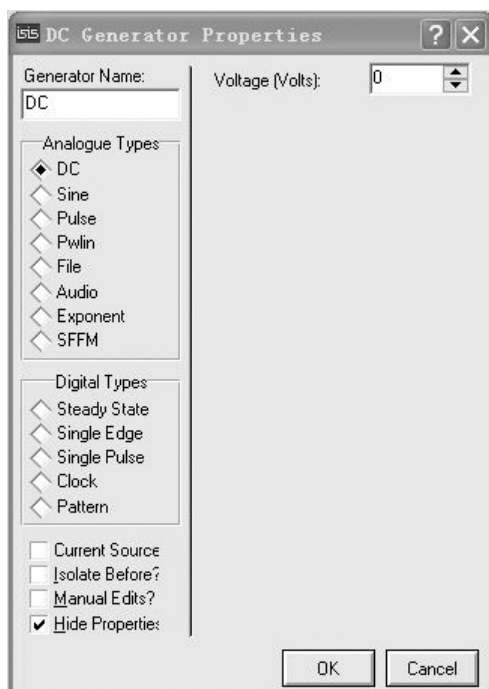


图 4-60 激励属性设置对话框

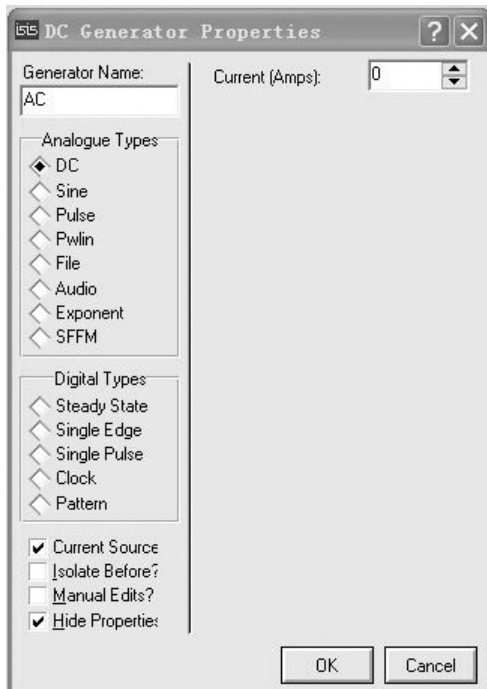


图 4-61 勾选选项

(4) 单击“OK”按钮，完成属性设置。

## 2) 正弦波信号发生器

(1) 放置正弦波信号发生器。

① 在 PROTEUS ISIS 环境下单击工具箱中的“Generator Mode”按钮图标，出现如图 4-58 所示的所有激励源名称列表。

② 用鼠标左键单击“SINE”，则在预览窗口出现正弦波信号发生器的符号。

③ 在编辑窗口双击，则正弦波信号发生器被放置到原理图编辑界面中，可使用镜像、翻转工具对其位置和方向进行调整。

(2) 编辑正弦波信号发生器。

① 双击原理图中的正弦波信号发生器符号，出现其属性设置对话框，如图 4-62 所示。

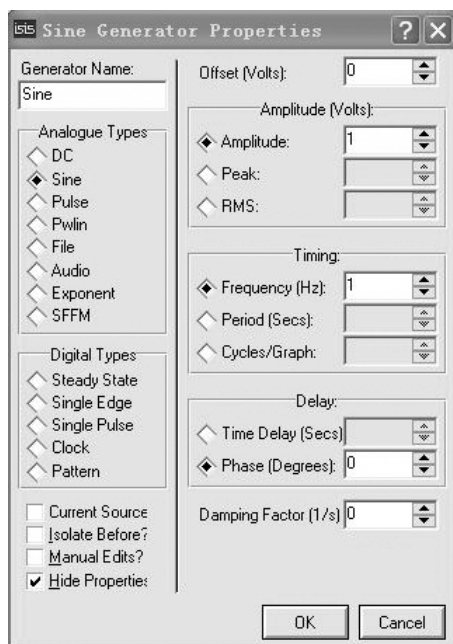


图 4-62 正弦波信号属性设置对话框

➤ Offset (Volts): 补偿电压，即正弦波的振荡中心电平。

➤ Amplitude (Volts): 正弦波的三种幅值标记方法，其中 Amplitude 为振幅，即半波峰值电压，Peak 为峰值电压，RMS 为有效值电压，以上三个电压值选填一项即可。

➤ Timing: 正弦波频率的三种定义方法，其中 Frequency (Hz) 为频率，单位为赫兹；Period (Secs) 为周期，单位为秒。这两项填一项即可。Cycles/Graph 为占空比，要单独设置。

➤ Delay: 延时，指正弦波的相位，有两个选项，选填一个即可。其中 Time Delay (Secs) 是时间轴的

延时，单位为秒；Phase (Degrees) 为相位，单位为度。

② 在“Generator Name”中输入正弦波信号发生器的名称，比如“SINE SOURCE 1”，在相应的项目中设置相应的值。

③ 单击“OK”按钮，完成设置。

## 2. 虚拟终端

PROTEUS VSM 提供的虚拟终端相当于键盘和屏幕的双重功能，免去了上位机系统的仿真模型，使用户在用到单片机与上位机之间的串行通信时，直接由虚拟终端经 RS-232 模型与单片机之间异步发送或接收数据。虚拟终端在运行仿真时会弹出一个仿真界面，当由 PC 向单片机发送数据时，可以和实际的键盘关联，用户可以从键盘经虚拟终端输入数据；当接收到单片机发送来的数据后，虚拟终端相当于一个显示屏，会显示相应信息。

调入虚拟终端：在 PROTEUS ISIS 环境下单击工具箱列表（见图 4-63）中的“VIRTUAL TERMINAL”按钮图标，出现如图 4-64 所示虚拟终端的原理图符号。

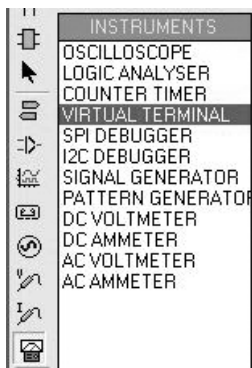


图 4-63 工具箱列表

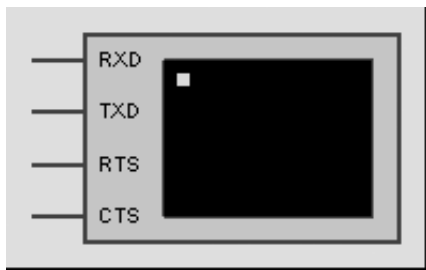


图 4-64 虚拟终端的原理图符号

虚拟终端共有 4 个接线端，其中 RXD 为数据接收端，TXD 为数据发送端，RTS 为请求发送信号，CTS 为清除传送，是对 RTS 的响应信号。

在使用虚拟终端时，首先要对其属性参数进行设置。双击元件，出现如图 4-65 所示的虚拟终端属性设置对话框，主要参数有下面几个。

- Baud Rate：波特率，范围为 300 ~ 57 600b/s。
- Data Bits：传输的数据位数，7 位或 8 位。
- Parity：奇偶校验位，包括奇校验、偶校验和无校验。
- Stop Bits：停止位，具有 0、1 或 2 位停止位。
- Send XON/XOFF：第 9 位发送允许/禁止。

选择合适参数后，单击“OK”按钮关闭对话框。运行仿真，弹出如图 4-66 所示的虚拟终端的仿真界面。

用户在图 4-66 所示的界面中可以看到从单片机发送来的数据，还能通过键盘把数据输入该界面，然后发送给单片机。

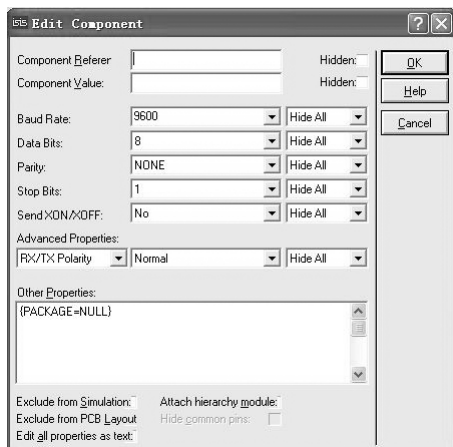


图 4-65 虚拟终端属性设置对话框



图 4-66 虚拟终端的仿真界面



## 4.8 库元件认识

### 1. 库元件的分类

PROTEUS ISIS 的库元件是按类存放的，即类→子类→元件。拾取元件的一般方法是按类查询。

1) 大类 (Category) 元件拾取对话框如图 4-67 所示。



图 4-67 元件拾取对话框

在左侧的“Category”中共列出了以下几个大类，其含义如表 4-5 所示。

表 4-5 大类列表

Category (类)	含 义	Category (类)	含 义
Analog ICs	模拟集成器件	PLDs & FPGAs	可编程逻辑器件和现场可编程门阵列
Capacitors	电容	Resistors	电阻
CMOS 4000 series	CMOS 4000 系列	Simulator Primitives	仿真源
Connectors	接头	Speakers & Sounders	扬声器和声响
Data Converters	数据转换器	Switcher & Relays	开关和继电器
Debugging Tools	调试工具	Switching Devices	开关器件
Diodes	二极管	Thermionic Valves	热离子真空管
ECL 10000 Series	ECL 10000 系列	Transducers	传感器
Electromechanical	电动机	Transistors	晶体管
Inductors	电感	TTL 74 Series	标准 TTL 系列
Laplace Primitives	拉普拉斯模型	TTL 74ALS Series	先进的低功耗肖特基 TTL 系列
Memory ICs	存储器芯片	TTL 74AS Series	先进的肖特基 TTL 系列
Microprocessor ICs	微处理器芯片	TTL 74F Series	快速 TTL 系列
Miscellaneous	混杂器件	TTL 74HC Series	高速 CMOS 系列
Modelling Primitives	建模源	TTL 74HCT Series	与 TTL 兼容的高速 CMOS 系列
Operational Amplifiers	运算放大器	TTL 74LS Series	低功耗肖特基 TTL 系列
Optoelectronics	光电器件	TTL 74S Series	肖特基 TTL 系列

当要从库中拾取一个元件时，首先要清楚它的分类是位于表 4-5 中的哪一类，然后在打开的元件拾取对话框中，选中“Category”中相应的大类。

**2) 子类 (Sub - category)** 选取元件所在的大类 (Category) 后，再选子类 (Sub - category)，也可以直接选生产厂家 (Manufacturer)，这样会在元件拾取对话框中间部分的查找结果 (Results) 中显示符合条件的元件列表。从中找到所需的元件，双击该元件名称，元件即被拾取到对象选择器中。如果要继续拾取其他元件，最好使用双击元件名称的方法，对话框不会关闭。如果只选取一个元件，可以单击元件名称后单击“OK”按钮，关闭对话框。

如果选取大类后，没有选取子类或生产厂家，则在元件拾取对话框中的查询结果中，会把此大类下的所有元件按元件名称首字母的升序排列出来。

## 2. 各子类介绍

**1) Analog ICs** 模拟集成器件共有 8 个子类，如表 4-6 所示。

表 4-6 模拟集成器件

子 类	含 义
Amplifier	放大器
Comparators	比较器
Display Drivers	显示驱动器
Filters	滤波器

续表

子 类	含 义
Miscellaneous	混杂器件
Regulators	三端稳压器
Timers	555 定时器
Voltage References	参考电压

2) Capacitors 电容共有 23 个分类, 如表 4-7 所示。

表 4-7 电容

子 类	含 义
Animatted	可显示充放电电荷电容
Audio Grade Axial	音响专用电容
Axial Lead polypropene	径向轴引线聚丙烯电容
Axial Lead polystyerne	径向轴引线聚苯乙烯电容
Ceramic Disc	陶瓷圆片电容
Decouplong Disc	解耦圆片电容
Generic	普通电容
High Temp Radial	高温径向电容
High Temp Sxial Electrolytic	高温径向电解电容
Metallised Polyester Film	金属聚酯膜电容
Metallised Polypropene	金属聚丙烯电容
Metallised Polypropene Film	金属聚丙烯膜电容
Miniture Electrolytic	微型电解电容
Multilayer Metallised Polyester Film	多层金属聚酯膜电容
Mylar Film	聚酯薄膜电容
Nickel Barrier	镍栅电容
Non Polarised	无极性电容
Polyester Layer	聚酯层电容
Radial Electrolytic	径向电解电容
Resin Dipped	树脂蚀刻电容
Tantalum Bead	钽珠电容
Variable	可变电容
VX Axial Electrolytic	VX 轴电解电容

3) CMOS 4000 series CMOS 4000 系列数字电路共有 16 个分类, 如表 4-8 所示。

表 4-8 CMOS 4000 系列数字电路

子 类	含 义
Adders	加法器
Buffers & Drivers	缓冲和驱动器

续表

子 类	含 义
Comparators	比较器
Counters	计数器
Decoders	译码器
Encoders	编码器
Flip – Flops & Latches	触发器和锁存器
Frequency Dividers & Tiner	分频和定时器
Gates & Inverters	门电路和反相器
Memory	存储器
Misc. Logic	混杂逻辑电路
Multiplexers	数据选择器
Multivebrators	多谐振荡器
Phase – locked Loops ( PLL)	锁相环
Registers	寄存器
Signal Switcher	信号开关

4) Connectors 接插件共有 9 个分类，如表 4-9 所示。

表 4-9 接插件

子 类	含 义
Sudio	音频接头
D – type	D 型接头
DIL	双排插座
Header Blocks	插头
Miscellaneous	各种插头
PCB Transfer	PCB 传输接头
SIL	单排插座
Ribbo Cable	蛇皮电缆
Terminal Blocks	接线端子台

5) Data Converters 数据转换器共有 4 个分类，如表 4-10 所示。

表 4-10 数据转换器

子 类	含 义
A/D Converters	模数转换器
D/AConverters	数模转换器
Sample & Hold	采样保持器
Temperature Sensors	温度传感器

6) Debugging Tools 调试工具共有 3 个分类，如表 4-11 所示。

表 4-11 调试工具

子 类	含 义
Breakpoint Triggers	断电触发器
Logic Probes	逻辑输出探针
Logic Stimuli	逻辑状态输入

7) Diodes 二极管共有 8 个分类，如表 4-12 所示。

表 4-12 二极管

子 类	含 义
Bridge Rectifiers	整流桥
Generic	普通二极管
Rectifiers	整流二极管
Schottky	肖特基二极管
Swicing	开关二极管
Tunnel	隧道二极管
Varicap	变容二极管
Zener	稳压二极管

8) Inductors 电感共有 3 个分类，如表 4-13 所示。

表 4-13 电感

子 类	含 义
Generic	普通电感
SMT Inductors	表面安装技术电感
Transformers	电压器

9) Laplace Primitives 拉普拉斯模型共有 7 个分类，如表 4-14 所示。

表 4-14 拉普拉斯模型

子 类	含 义
1st Order	一阶模型
2nd Order	二阶模型
Controllers	控制器
Non - Linear	非线性模型
Operators	算子
Poles/Zeros	极点/零点
Symbols	符号

10) Memory ICs 存储器芯片共有 7 个分类，如表 4-15 所示。

表 4-15 存储器芯片

子 类	含 义
Dynamic RAM	动态数据存储器
EEPROM	点可擦除程序存储器
EPROM	可擦除程序存储器
I <sup>2</sup> C Memories	I <sup>2</sup> C 总线存储器
Memory Cards	存储卡
SPI Memories	SPI 总线存储器
Static RAM	静态数据存储器

11) Microprocessor ICs 微处理器芯片共有 13 个分类, 如表 4-16 所示。

表 4-16 微处理器芯片

子 类	含 义
68000 Family	68000 系列
8051 Family	8051 系列
ARM Family	ARM 系列
AVR Family	AVR 系列
BASIC Stamp Modules	Parallax 公司微处理器
HC11 Family	HC11 系列
Peripherals	CPU 外设
PIC10 Family	PIC10 系列
PIC12 Family	PIC12 系列
PIC16 Family	PIC16 系列
PIC18 Family	PIC18 系列
PIC24 Family	PIC24 系列
Z80 Family	Z80 系列

12) Modelling Primitives 建模源共有 9 个分类, 如表 4-17 所示。

表 4-17 建模源

子 类	含 义
Analog (SPOCE)	模拟 (仿真分析)
Digital (Buffers & Gates)	数字 (缓冲器和门电路)
Digital (Combinational)	数字 (组合电路)
Digital (Miscellaneous)	数字 (混杂)
Digital (Sequential)	数字 (时序电路)
Mixed Mode	混合模式
PLD Elements	可编程逻辑器件单元
Realtime (Actuators)	实时激励器
Realtime (Indictors)	实时指示器

13) Operational Amplifiers 运算放大器共有 7 个分类, 如表 4-18 所示。



表 4-18 运算放大器

子 类	含 义
Dual	双运放
Ideal	理想运放
Macromidel	大量使用运放
Octal	八运放
Quad	四运放
Single	单运放
Triple	三运放

14) Optoelectronics 光电器件共有 11 个分类, 如表 4-19 所示。

表 4-19 光电器件

子 类	含 义
7 - Segment Displays	7 段显示
Slphanumeric LCDs	字符液晶显示器
Bargraph Displays	条形显示
DOT Matix Displays	点阵显示
Graphical LCDs	图形液晶显示器
Lamps	灯
LCD Controllers	液晶控制器
LCD Panels Displays	液晶面板显示
LEDs	发光二极管
Optocouplers	光电耦合
Serial LCDs	串行口液晶显示器

15) Resistors 电阻共有 11 个分类, 如表 4-20 所示。

表 4-20 电阻

子 类	含 义	子 类	含 义
0.6W Metal Film	0.6W 金属膜电阻	High	高压电阻
10W Wirewound	10W 绕线电阻	NTC	负温度系数热敏电阻
2W MetalFilm	2W 金属膜电阻	Resistor	排阻
3W Wirewound	3W 绕线电阻	Variable	滑动变阻器
7W Wirewound	7W 绕线电阻	Varisitors	可变电阻
Generic	普通电阻		

16) Simulator Primitives 仿真源共有 3 个分类, 如表 4-21 所示。

表 4-21 仿真源

子 类	含 义
Flip - Flops	触发器

续表

子 类	含 义
Gates	门电路
Sources	电源

17) Switches and Relays 开关和继电器共有 4 个分类, 如表 4-22 所示。

表 4-22 开关和继电器

子 类	含 义
Key pads	键盘
Relays ( Generic )	普通继电器
Relays ( Specific )	专用继电器
Switches	开关

18) Switching Devices 开关器件共有 4 个分类, 如表 4-23 所示。

表 4-23 开关器件

子 类	含 义
DIACs	两端交流开关
Generic	普通开关元件
SCRs	可控硅
TRIACs	三端双向可控硅

19) Thermionic Valves 热离子真空管共有 4 个分类, 如表 4-24 所示。

表 4-24 热离子真空管

子 类	含 义
Diodes	二极管
Pentodes	五极真空管
Teterodes	四极管
Triodes	三极管

20) Transducers 传感器共有两个分类, 如表 4-25 所示。

表 4-25 传感器

子 类	含 义
Pressure	压力传感器
Temperature	温度传感器

21) Transistors 晶体管共有 8 个分类, 如表 4-26 所示。

表 4-26 晶体管

子 类	含 义
Bipolar	双极型晶体管

续表

子 类	含 义
Generic	普通晶体管
IGBT	绝缘双极型晶体管
JFET	结型场效应管
MOSFET	金属氧化物场效应管
RF Power LDMOS	射频功率 LDMOS 管
RF Power VDMOS	射频功率 VDMOS 管
Unijunction	单结晶体管



## 4.9 思考与练习

1. 熟悉 PROTEUS 的软件环境。
2. 基于 PROTEUS 的产品设计流程具体是什么？
3. 创建一个空白原理图文件，要求：① 图纸大小设置为 A4，图纸方向设置为横向；② 图纸边沿设置为黑色，图纸颜色设置成蓝色；③ 图纸字体设置为常规，字体大小为 18 号；④ 工作面网格形状设置为点状网格；⑤ 设置文件名为 MyFirst.DSN 并保存。
4. 如何把图形编辑窗口的背景色改成白色？
5. 如何向图形编辑窗口添加单片机 AT89C51？
6. 创建米字形数码管的动态模型。
7. 创建 555 定时器的原理图模型。
8. 虚拟终端的作用是什么？如何应用？

# 结 合 篇

# 第 5 章 PROTEUS ISIS 电路仿真

PROTEUS ISIS 能够仿真模拟电路、数字电路、模数混合电路，包括所有相关器件的基于微处理器设计的协同仿真。PROTEUS ISIS 中的整个电路分析是在 ISIS 原理图设计模块中延续下来的，原理图中电路激励、虚拟仪器、曲线图及直接布置在线路上的探针一起出现在电路中，任何时候都可以通过仿真控制按钮完成实时仿真。



## 5.1 交互式仿真

### 1. PROTEUS ISIS 交互式仿真控制面板

交互式电路仿真是电路分析的一个重要部分。输入原理图后，通过在期望的观察点放置电流/电压探针或虚拟仪器，再单击“运行”按钮，即可观察到电路的实时输出。

1) 仿真控制按钮 交互式仿真由一个貌似播放机操作按钮的控制面板控制，如图 5-1 所示。



图 5-1 仿真控制按钮

从左至右控制面板功能依次如下。

- 运行按钮：开始仿真。
- 步进按钮：此按钮可以使仿真按照预设的时间步长（单步执行时间增量）进行仿真。每单击一次，仿真进行一个步长事件后停止。若按下按键后不放开，仿真将连续进行，直到按停止键。步长的默认值为 50ms，可以对其进行设置。这一功能可更好地细化监控电路，同时也可以使电路放慢动作工作，从而更好地了解电路各元器件间的相互关系。
- 暂停按钮：暂停按钮可延续仿真的进行，再次按下可继续暂停的仿真，也可在暂停后接着进行步进仿真。
- 停止按钮：可使 PROTEUS 停止实时仿真，所有可动状态停止，模拟器不占用内存。除激励元器件、开关等外，所有指示器重置为停止时的状态。

### 2) 仿真设置

(1) 元器件引脚逻辑状态。在仿真过程中，连接在数字或混合网络的元器件引脚会显示一个有色小正方形，默认蓝色表示逻辑 0，红色表示逻辑 1，灰色表示不固定。以上 3 种颜色可通过选择【Template】→【Set Design Defaults】菜单项改变，设置界面如图 5-2 所示。

(2) 用不同颜色电路连线显示相应电压。在仿真过程中，可以用不同颜色电路连线显示相应电压，默认为蓝色表示 -6V，绿色表示 0V，红色表示 +6V。连线颜色按照从蓝到红的颜色深浅随电压由小到大的规律渐变。上述颜色可通过图 5-2 所示界面进行设置。若需

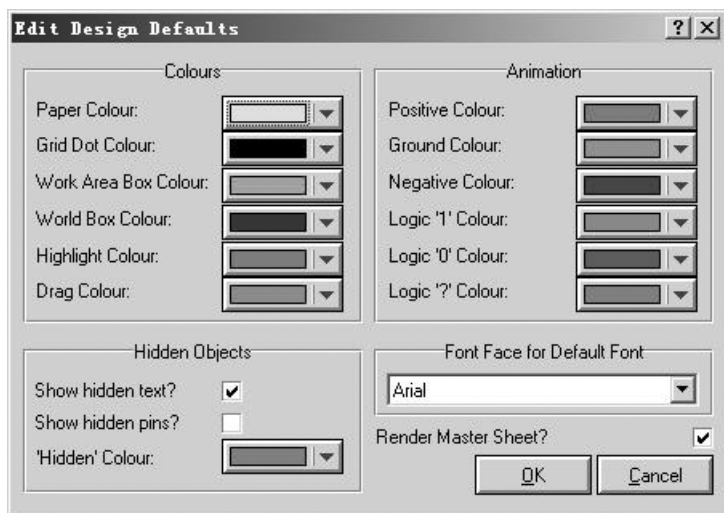


图 5-2 设置元器件引脚状态界面

改变电压的上下限值，可以选择【System】→【Set Animation Options】菜单项进行设置，如图 5-3 所示。

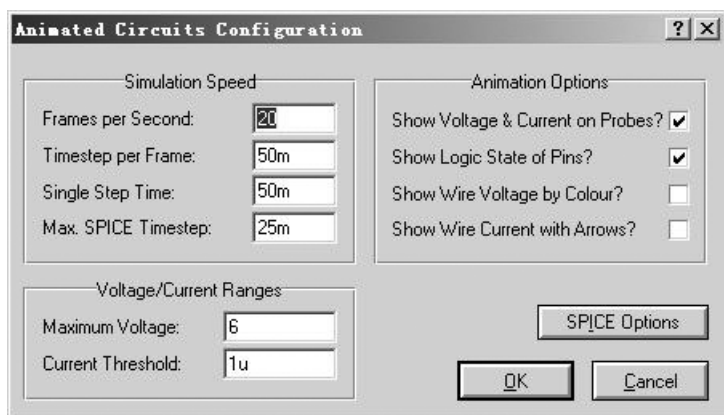


图 5-3 设置电压的上下限值

(3) 用箭头显示电流方向。此功能可使电路连线显示出电流的具体流向。应当注意，当线路电流小于设置的起始电流（默认值为  $1\mu\text{A}$ ）时，箭头不显示。起始电流可以在图 5-3 所示界面中修改。

(4) 设置仿真帧频及每帧仿真时间。帧频即每秒屏幕更新次数，一般取默认值即可，但有时在调试过程中可适当减小。每帧仿真时间可使电路运行更慢或更快，必要时可根据具体需要更改数值，通过图 5-3 所示界面进行设置。

在交互式电路仿真过程中，需注意以下几点。

- 运行时间方面：在增加每帧仿真时间时应保证 CPU 能够实现。另外，模拟分析要比数字分析慢得多。
- 电压范围：如果想用连线颜色来显示节点电压，则须预先估计电路中可能出现的电压范围。因为默认范围仅为  $-6 \sim +6\text{V}$ ，因此，必要时可重新设置。
- 高阻抗点：电路中若有未连接处，系统仿真时自动加入高阻抗电阻代替，而不会提示连线错误，所

以将产生错误结果而不容易被发现，连线时应特别注意。

## 2. PROTEUS ISIS 交互式仿真活性元件

除了一些通用元件外，PROTEUS ISIS 交互式仿真通常使用一些活性元件进行电路仿真，如图 5-4 所示。

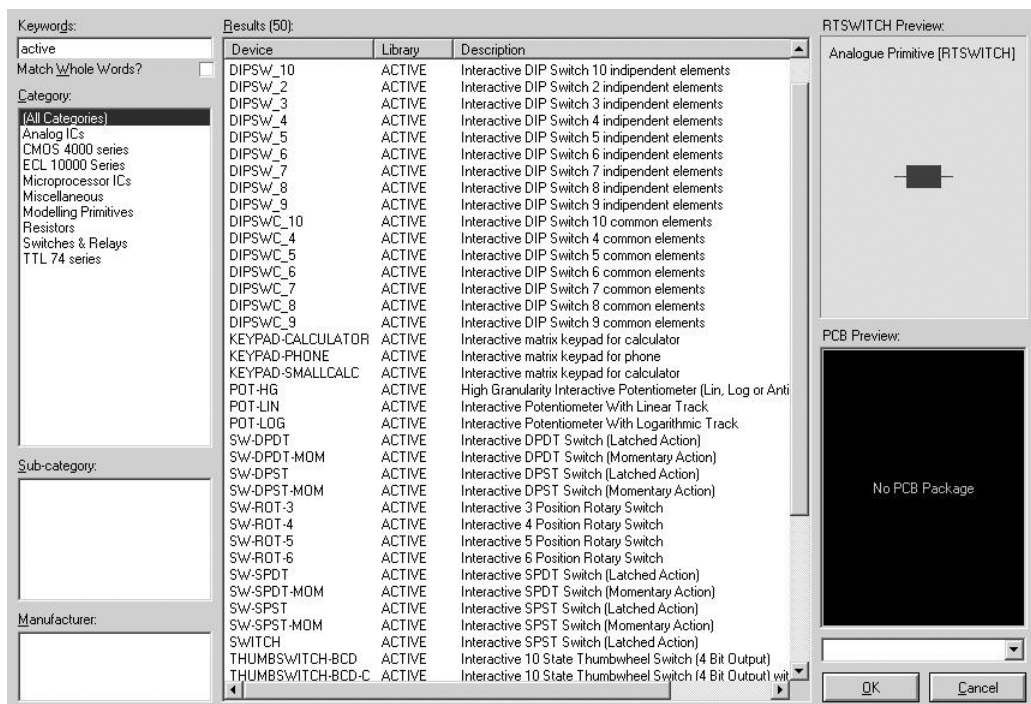


图 5-4 活性元件

活性元件具有指示结构及操作结构，如图 5-5 所示。

指示结构以图形状态显示其在电路中的状态，图 5-5 中的开关处于打开状态。

操作结构为红色的标记，单击相应的标记，活性元件就会执行相应的操作。单击图 5-5 中的 ⬇️ 标记，开关关闭，如图 5-6 所示。



图 5-5 活性元件




图 5-6 单击 ⬆️ 标记开关闭合

## 3. PROTEUS ISIS 交互式仿真过程

下面以图 5-7 所示电路为例说明 PROTEUS ISIS 交互式仿真的过程。

PROTEUS ISIS 交互式仿真过程如下。

(1) 点选  (Component) 图表，单击“P”按钮，从弹出的选取元件对话框中选择仿真元件。滑动电阻仿真元件选取如图 5-8 所示。

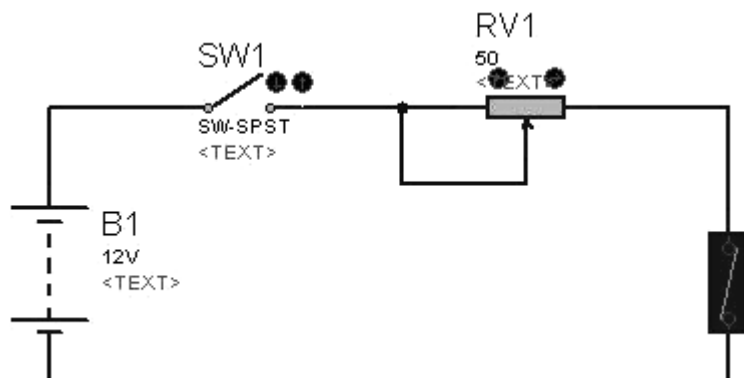


图 5-7 交互式仿真电路

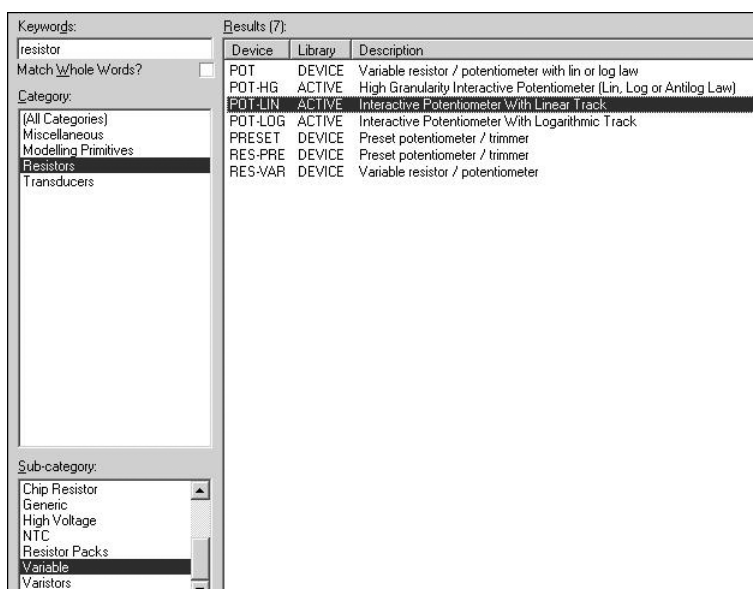


图 5-8 选择滑动电阻仿真模型

要求所选择的元件具有仿真模型，滑动电阻仿真模型如图 5-9 所示。

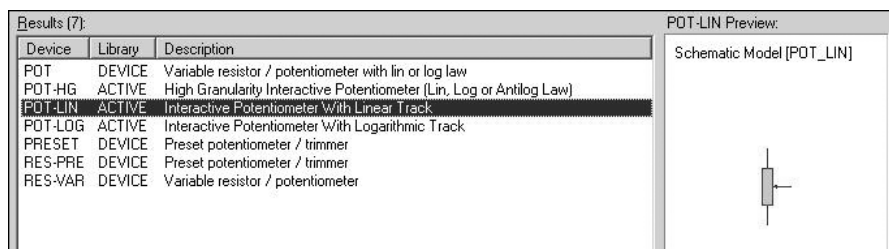


图 5-9 选择具有仿真模型的元件

双击元件名，添加元件到对象选择器中。

按照上述放置方法，添加电源仿真元件，如图 5-10 所示。



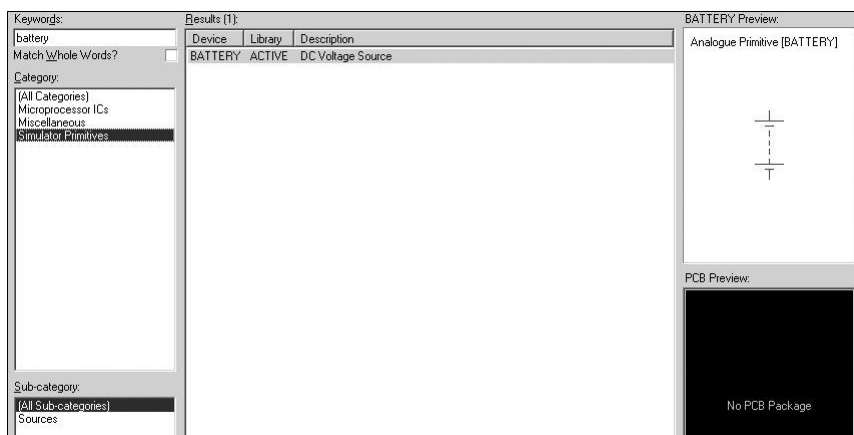


图 5-10 添加电源仿真元件

添加熔丝仿真元件, 如图 5-11 所示。

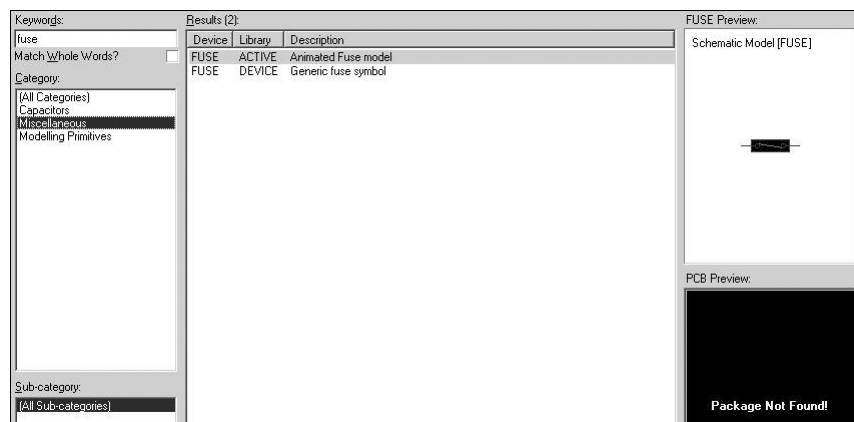
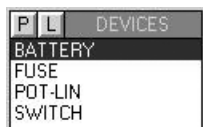
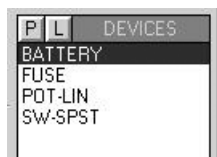
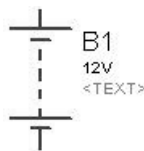
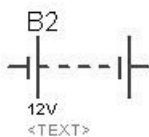


图 5-11 添加熔丝仿真元件

此时对象选择器中将列出所有元件, 如图 5-12 所示。

(2) 从对象选择器中选择相应的元件, 在原理图编辑窗口单击, 此时系统处于放置模式。移动鼠标, 元件将随着鼠标的移动而移动, 如图 5-13 所示。

(3) 在希望放置元件的位置单击鼠标左键放置元件, 如图 5-14 所示。

图 5-12 对象选择器中  
列出所有已选择元件图 5-13 添加元  
件到编辑窗口图 5-14 在原理图  
编辑窗口放置电源

按照上述方法放置其他元件, 并按照图 5-7 所示布局布置元件, 结果如图 5-15 所示。

(4) 双击电源元件, 打开元件属性编辑对话框, 如图 5-16 所示。

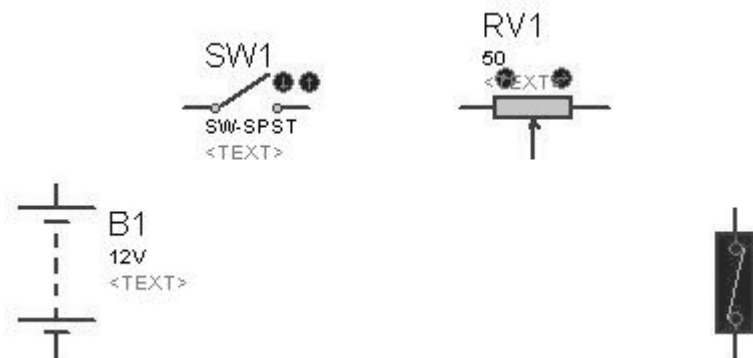


图 5-15 布置元件

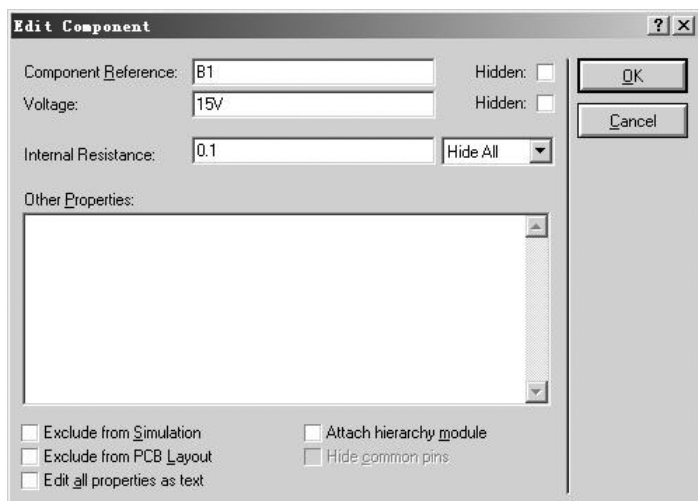


图 5-16 电源元件属性编辑对话框

设置元件属性，单击“OK”按钮完成设置，结果如图 5-17 所示。

其他元件属性值参照电路显示（图 5-7 所示），按照上述方法进行设置，设置结果如图 5-18 所示。

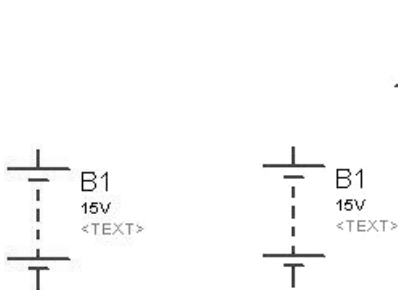


图 5-17 电源属性

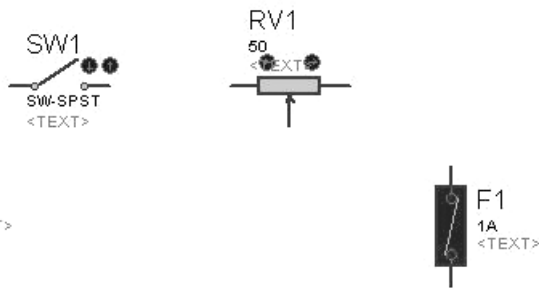


图 5-18 电路元件属性

(5) 按图 5-7 所示连接电路。将光标放置到元件连接点，光标将以绿色笔状出现，如图 5-19 所示。

单击鼠标左键，开始画线，如图 5-20 所示。

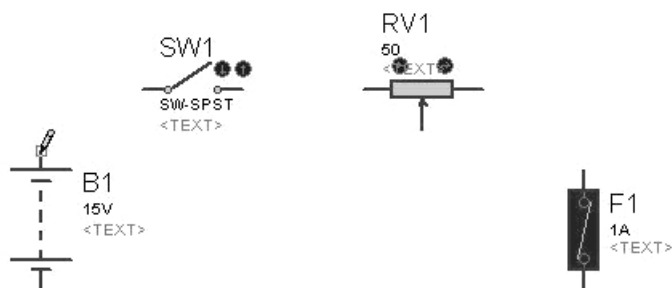


图 5-19 在元件连接点光标以绿色笔状出现

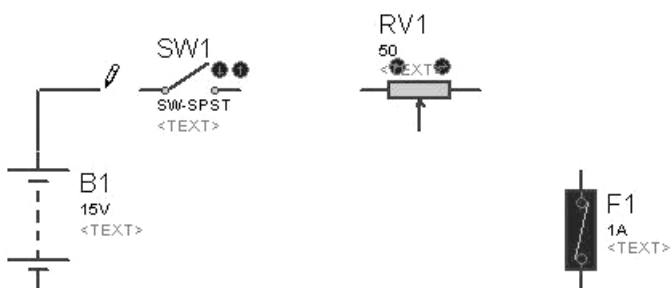


图 5-20 画线

在线的结束点，光标再次以绿色笔状出现，单击鼠标左键，结束画线，结果如图 5-21 所示。

按照上述方式，参照图 5-7 连接电路，连接好的电路如图 5-22 所示。

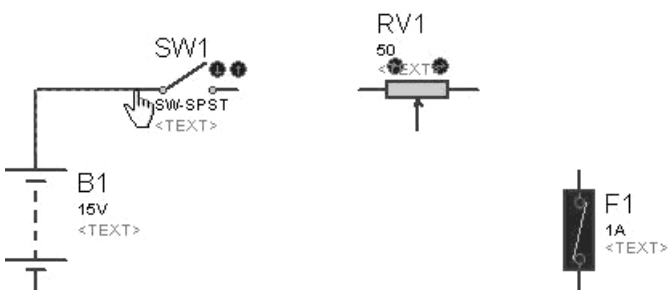


图 5-21 放置线结束点

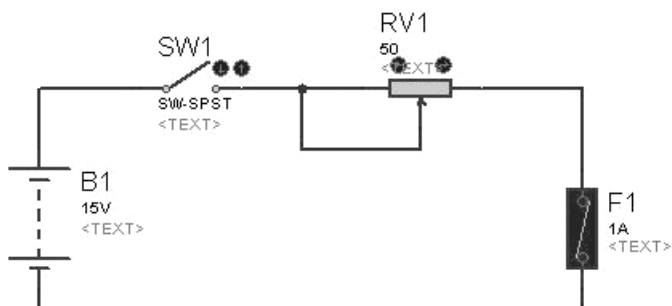


图 5-22 连接好的电路

(6) 单击控制面板中的“运行”按钮运行电路，电路运行结果如图 5-23 所示。

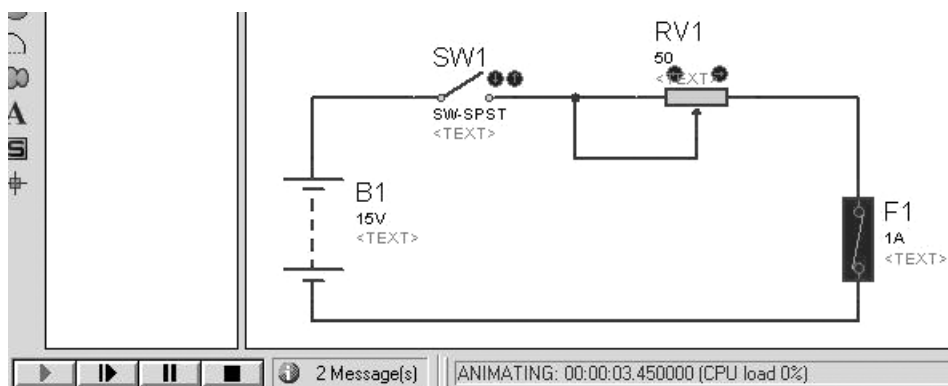


图 5-23 电路运行结果

在 PROTEUS ISIS 中给出仿真信息、仿真时间及 CPU 加载率。

用鼠标左键单击仿真信息，将弹出仿真日志窗口，如图 5-24 所示。



图 5-24 仿真日志

单击电路中标记的开关，闭合电路，此时电路的仿真结果如图 5-25 所示。

单击滑动变阻器的 ➡，减小电路中的电阻，此时电路仿真结果如图 5-26 所示。从仿真结果可知，熔丝开始变红。当继续减小电路中的电阻值时，熔丝熔断，如图 5-27 所示。单击滑动变阻器的 ⬅，增大电路中的电阻，电路仿真结果如图 5-28 所示。

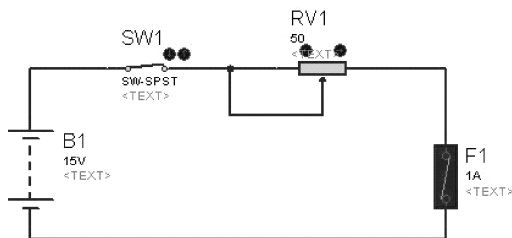


图 5-25 闭合开关电路仿真结果

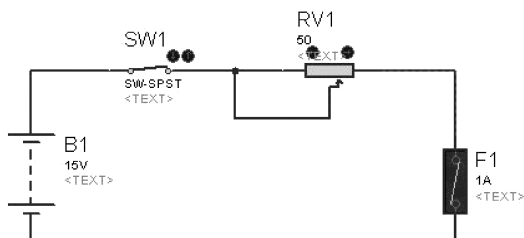


图 5-26 减小电路中电阻的仿真结果

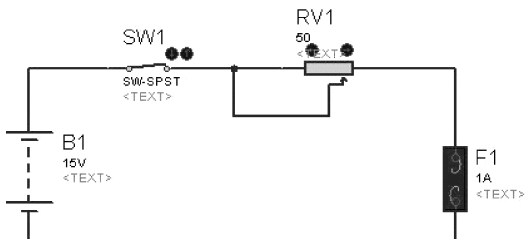


图 5-27 继续减小电路中的电阻值、熔丝熔断

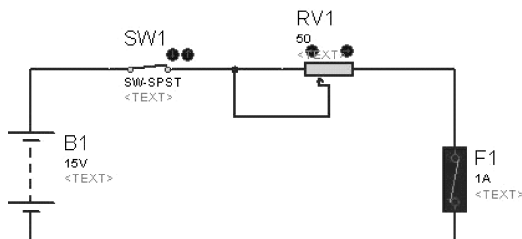


图 5-28 增大电路中电阻的仿真结果

单击“停止”按钮结束仿真。再次单击仿真信息，弹出仿真日志如图 5-29 所示。

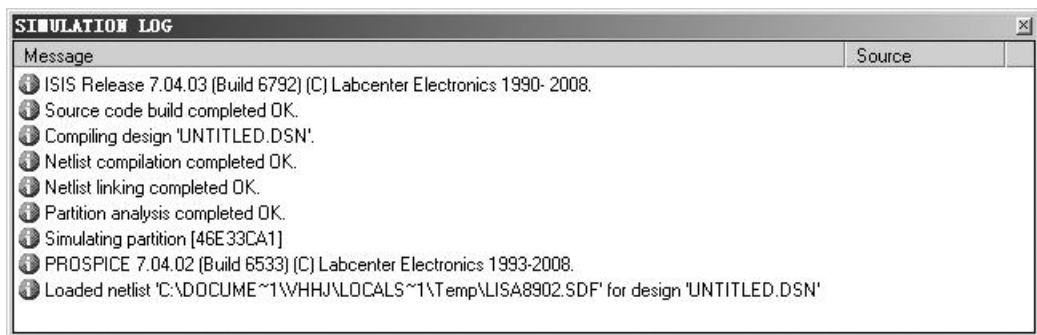


图 5-29 仿真结束后的仿真日志信息



## 5.2 基于图表的仿真


PROTEUS VSM 的虚拟仪器为用户提供了交互动态仿真功能，但这些仪器的仿真结果和状态会随着仿真的结束而结束，不能满足打印及长时间的分析要求。所以 PROTEUS ISIS 还提供了一种静态的图表仿真功能，无须运行仿真，随着电路参数的修改，电路中的各点波形将重新生成，并以图表的形式留在电路图中，供以后分析和打印。

图表仿真设计一系列按钮和菜单的选择，主要目的是把电路中某点对地的电压或某条支路的电流相对时间轴的波形自动绘制出来。图表仿真功能的实现包含以下步骤：

- ① 在电路中的被测点添加电压探针，或在被测支路添加电流探针。
- ② 选择放置波形的类别，并在原理图中拖出用于生成仿真波形的图标框。
- ③ 在图标框中添加探针。
- ④ 设置图表属性。
- ⑤ 单击图表仿真按钮生成所加探针对应的波形。
- ⑥ 存盘并打印输出。

下面具体介绍。

**1) 设置探针** 绘制好一个完整的电路，如图 5-30 所示。现在 we 希望在图中绘制电阻 R1 左端和 741 输出 6 端的电压波形。

首先需要在这两个点放置两个电压探针。在 PROTEUS ISIS 的左侧工具箱中选择电压探针（Voltage Probe）的按钮图标，在图 5-30 中的相应位置双击两次，放置两个电压探针。然后把电压探针与被测电压点连接在一起。

下面为两个电压探针命名。双击 R1 的电压探针，打开如图 5-31 所示的属性设置对话框。把探针命名为“INPUT”，单击“OK”按钮关闭对话框。另一个探针命名为“OUTPUT”。

**2) 设置波形类别** 在 PROTEUS ISIS 的左侧工具箱中选择图形模式（Graph Mode）的按钮图标，在对象选择区列出了所有的波形类别，如图 5-32 所示，其含义如表 5-1 所示。

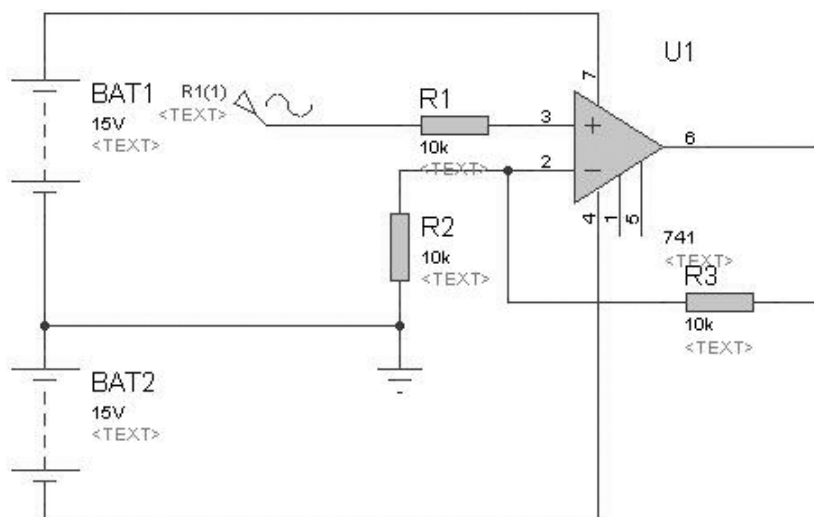


图 5-30 图表仿真电路举例

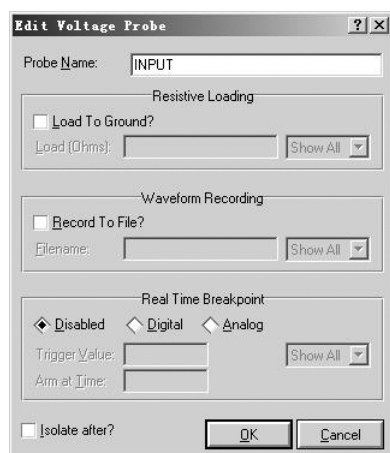


图 5-31 电压探针属性设置对话框

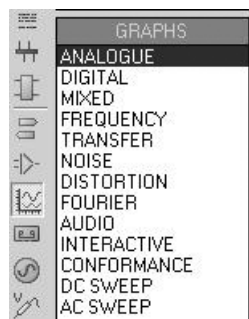


图 5-32 仿真波形类别

表 5-1 仿真波形类别含义

波形类别名称	含 义
ANALOGUE	模拟波形
DIGITAL	数字波形
MIXED	模数混合波形
FREQUENCY	频率响应
TRANSFER	转移特性分析
NOISE	噪声波形
DISTORTION	失真分析
FOURIER	傅里叶分析
AUDIO	音频分析
INTERACTIVE	交互分析

续表

波形类别名称	含 义
CONFORMANCE	一致性分析
DC SWEEP	直流扫描
AC SWEEP	交流扫描

在本例中,“INPUT”和“OUTPUT”都是模拟波形,故选定“ANALOGUE”模拟波形。用鼠标单击图 5-32 中的“ANALOGUE”,然后在原理图编辑区用鼠标左键拖出一个方框,如图 5-33 所示。

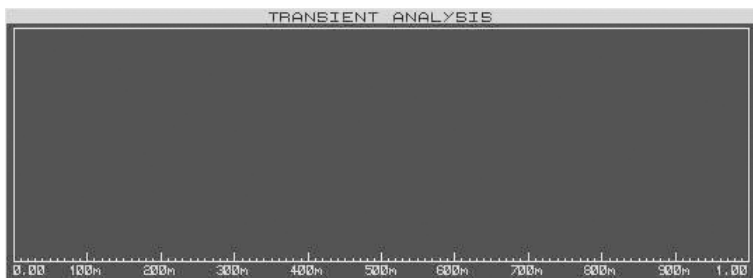


图 5-33 拖出的图标框

3) 添加探针 接下来在图标框中添加两个电压探针。选择主菜单【Graph】→【Add Trace】,如图 5-34 所示,打开如图 5-35 所示的添加轨迹对话框。

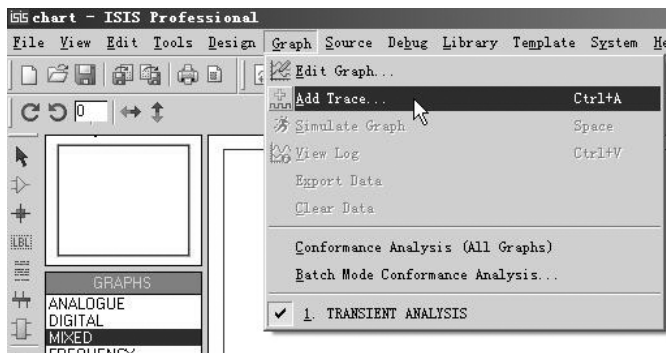


图 5-34 选择【Add Trace】命令

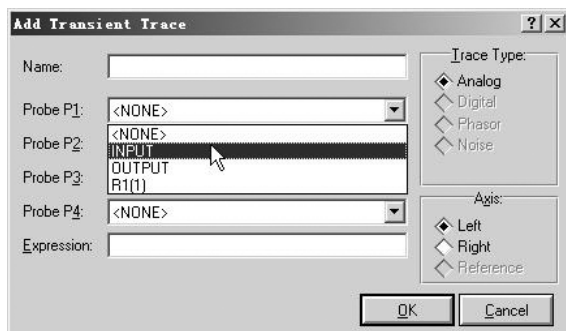


图 5-35 添加轨迹对话框

在图 5-35 中,单击轨迹类型 (Trace Type) 下面的“Analog”选择模拟波形,单击“Probe P1”的下拉箭头,出现如图 5-35 所示的所有探针名称。选中“INPUT”,则该探针自动添加到“Name”栏中。

接下来添加探针“OUTPUT”,重新选择主菜单【Graph】→【Add Trace】,打开如图 5-35 所示的添加轨迹对话框。一切完成后,出现如图 5-36 所示的图标框,图中多出了刚添加的两个探针的名称。

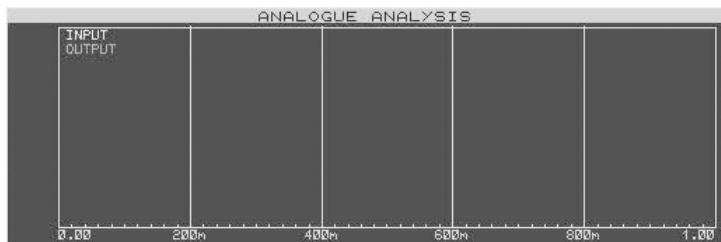


图 5-36 添加探针后的图标框

4) 设置图标属性 按空格键或选择【Graph】→【Simulate Graph】命令,生成波形,如图 5-37 所示。可以看到,输出信号与输入信号为同相位、同频率信号,但图形只是一个周期的波形。这是因为图标框的时间轴太短,默认为 1s。接下来修改波形的时间轴。

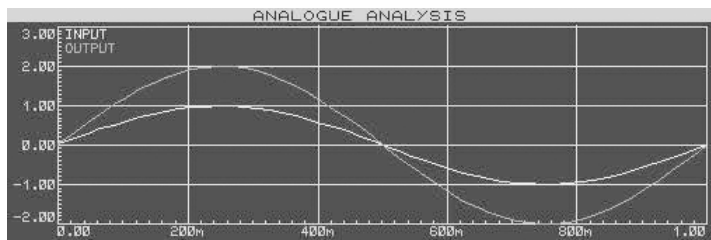


图 5-37 生成一个周期波形

双击图标框,打开如图 5-38 所示的对话框,把“Stop time”改为 6s,因为我们设计的电路中,波形的周期为 1s,这样可以显示 6 个周期的波形。同时在图中的“Graph title”栏可以修改或设置图标的名称,默认名称为“TRANSIENT ANALYSIS”暂态分析。

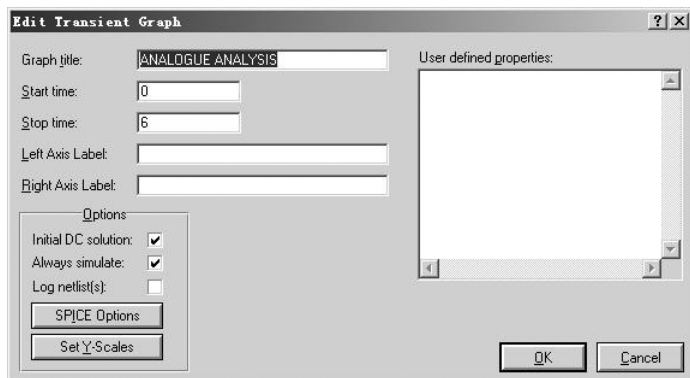


图 5-38 编辑图形属性对话框



将鼠标指向原理图中的图标框名称“TRANSIENT ANALYSIS”，在绿色区双击，会出现如图 5-39 所示的对话框，可以设定背景及图形颜色等。

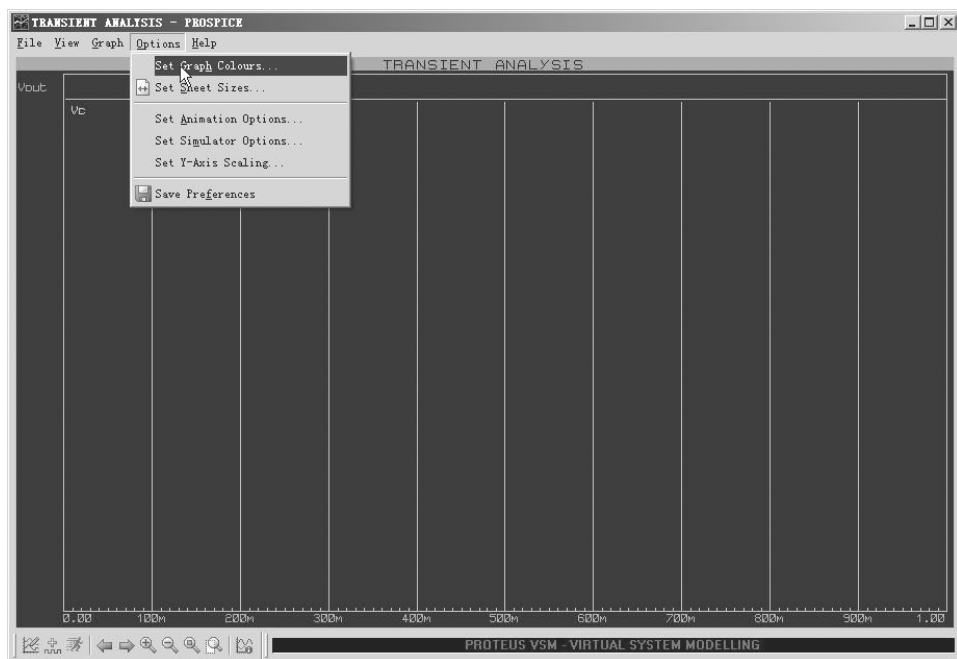


图 5-39 图标框的属性设置对话框

5) 生成波形 按空格键或选择【Graph】→【Simulate Graph】命令重新生成波形，如图 5-40 所示。

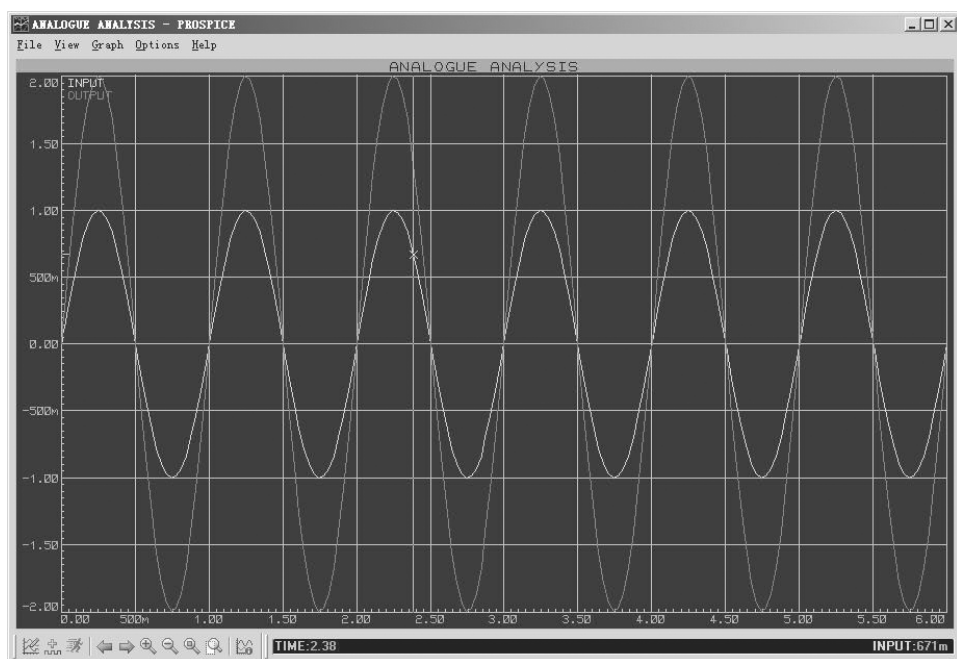


图 5-40 6 个周期波形

在波形中输入信号的峰值点附近单击鼠标左键，将出现测量指针，指针测量点如图 5-41 所示。

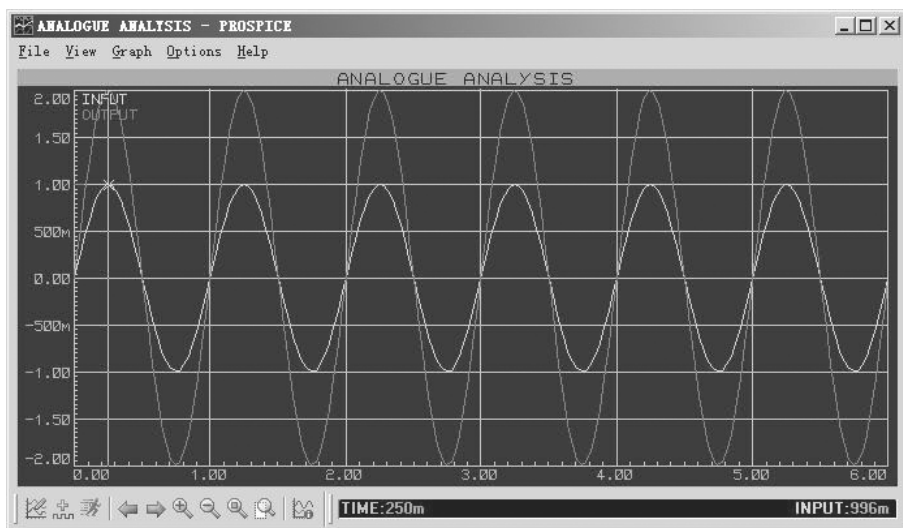


图 5-41 模拟图标测量指针测量输入信号

从图中的测量结果可知，输入信号在 250ms 处的电压值为 996mV。按下“Ctrl”键，在图标中输出信号的峰值点附近单击鼠标左键，将出现另一个测量指针，如图 5-42 所示。

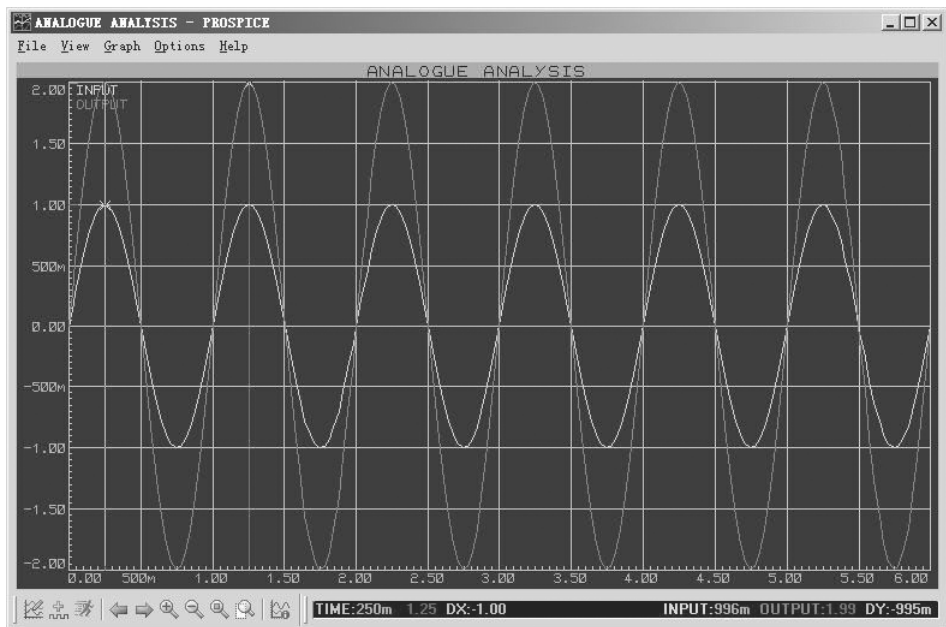


图 5-42 模拟图标测量指针测量输出信号

从图中的测量结果可知，输出信号在 1.25s 处的电压值为 1.99V。系统的仿真结果与理论计算结果相符。



### 5.3 思考与练习

1. 根据本章知识，完成图 5-43 所示电平指示电路的绘制和仿真。

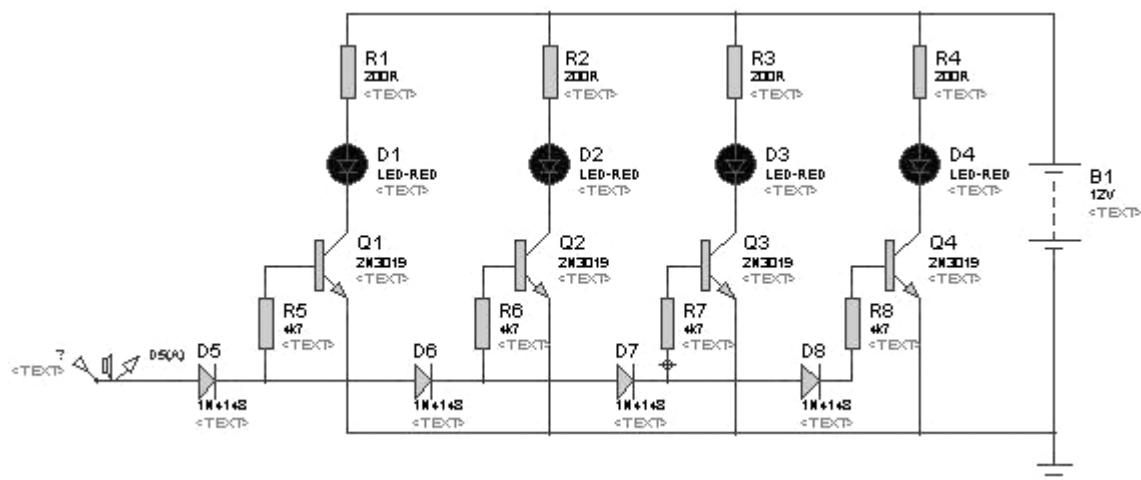


图 5-43 习题 5-1 图

2. 熟悉交互式仿真中的活性元件。

# 第6章 基于 PROTEUS ISIS 的 电工基础的电路仿真

欧姆定律、基尔霍夫定律、戴维南定理和诺顿定理是电学中基本的规律和原理。欧姆定律反应了电路中各种量的关系，基尔霍夫定律和戴维南定理用于分析和测量电路中各支路电流和各元器件上的电压。本章基于 PROTEUS 设计与仿真平台，设计电路验证各定律和定理的正确性。



## 6.1 基于 PROTEUS ISIS 的直流电路欧姆定律的设计与仿真

欧姆定律是电工学中的一条基本规律，它反映了电路中电压、电流、电阻三个重要参数之间的定量关系。本实验是利用 ROTEUS ISIS 设计与仿真平台，通过改变电路参数测试直流电阻的伏安特性，来验证欧姆定律。

欧姆定律的基本内容：流过一段电路的电流与这段电路两端的电压成正比，与这段电路的电阻成反比。用公式表示为：

$$R = U/I \quad U = IR \quad I = U/R$$

### 1. PROTEUS ISIS 电路设计


(1) 从 PROTEUS 库中选取元器件，如表 6-1 所示。

表 6-1 元器件明细表

元器件名称	类 别	子 类	标 识	值
RES/ $\Omega$	DEVICE	Generic	R	10
POT - LIN/ $\Omega$	ACTIVE	Rectifiers	RP	100
SWITCH	ACTIVE	Switch	SW	—
BATTERY/V	ACTIVE	SOURCES	B	10

(2) 放置元器件、连线（如图 6-1 所示），右击元器件进行属性设置，完成电气检测。

### 2. PROTEUS ISIS 调试与仿真

按图 6-1 所示连接好电路，断开开关，单击仿真按钮开启仿真，如图 6-2 所示，电压表、电流表显示数值。

合上开关，调节变阻器，观察并记录电压表与电流表的数值变化，求出电阻阻值，填入表 6-2。

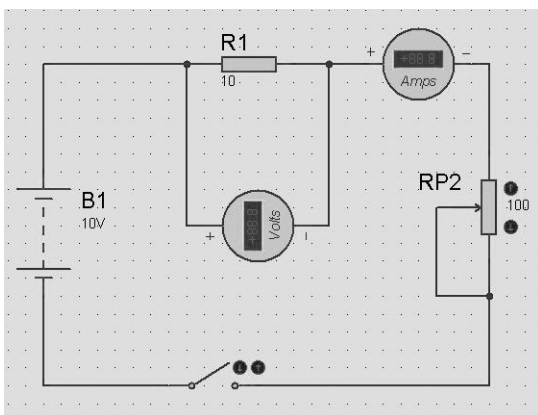


图 6-1 欧姆定律电路设计

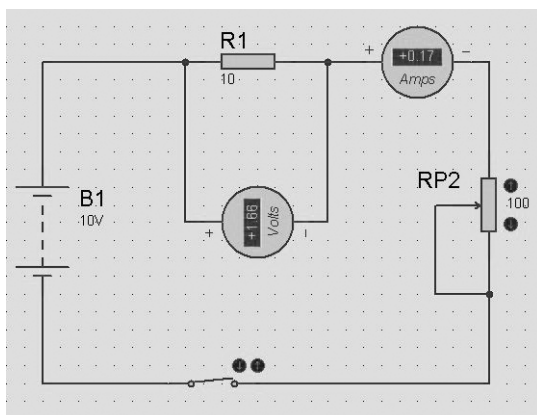


图 6-2 欧姆定律仿真

表 6-2 测量数值

测量序号	1	2	3	4	5	6	7	8
$U/V$	1.00	1.11	1.25	1.43	1.66	2.00	2.49	3.32
$I/A$	0.10	0.11	0.12	0.14	0.17	0.20	0.25	0.33
$R_i/\Omega$	10.00	10.09	10.42	10.21	9.76	10.00	9.96	10.06

由表 6-2 可以得出：

$$R = \sum R_i / 8 = 10.16 \Omega$$

即

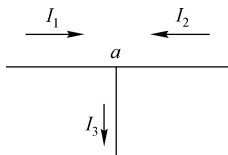
$$R \approx \sum (U/I)$$



## 6.2 基于 PROTEUS ISIS 的基尔霍夫定律的设计与仿真

### 1. 基尔霍夫电流定律

基尔霍夫电流定律是用来确定连接在同一节点上的各支路的电流关系的。由于电流的连续性，电路中的任何一点（包括节点在内）均不能堆积电荷。因此，在任一瞬时，流向某一节点的电流之和应该等于由该节点流出的电流之和。

图 6-3 节点  $a$  处的电流

如图 6-3 所示的电路中，对于节点  $a$  可以写出：

$$I_1 + I_2 = I_3$$

或将上式改写成：

$$I_1 + I_2 - I_3 = 0$$

$$\sum I = 0$$

即

#### 1) PROTEUS ISIS 电路设计

(1) 从 PROTEUS 库中选取元器件，如表 6-3 所示。

表 6-3 元器件明细表

元器件名称	类 别	子 类	标 识	值
BATTERY/V	ACTIVE	SOURCES	B1	12
RES/ $\Omega$	DEVICE	GENERIC	R1、R2、R3	100、150、300

(2) 根据图 6-4 所示连接电路, 设置各元器件属性, 并进行电气测量 (注意电压表与电流表的方向性)。

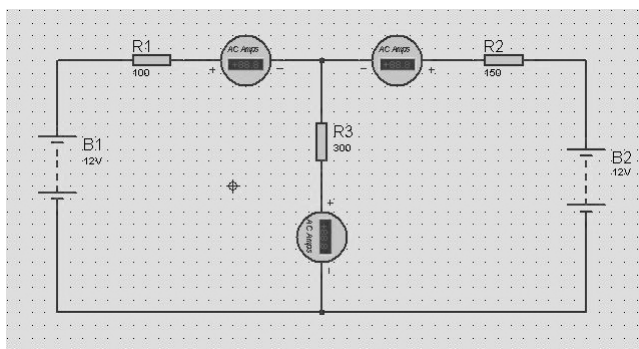


图 6-4 基尔霍夫电流定律连线图

2) PROTEUS ISIS 调试与仿真 如图 6-5 所示, 将电流表接入电路, 单击仿真按钮进行仿真, 记录流经节点  $b$  的电流数据。读出仿真电路图中各电流表的示数, R1 的电流  $I_1$  为 0.02A、R2 的电流  $I_2$  为 0.01A、R3 的电流  $I_3$  为 0.03A, 方向如图所示。

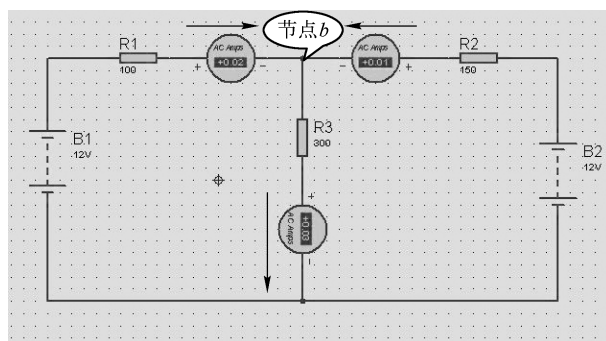


图 6-5 基尔霍夫电流定律调试与仿真

由图中可以看出  $I_1$ 、 $I_2$  为流入节点  $b$  的电流,  $I_3$  为流出节点  $b$  的电流, 由此可以得出  $I_1 + I_2 = I_3$ , 验证了电流定律的正确性。

## 2. 基尔霍夫电压定律

基尔霍夫电压定律是用来确定回路中各段电压之间关系的。如果从回路中的任意一点出发, 以顺时针方向或逆时针方向沿回路循环一周, 则在这个方向上的电位降之和应该等于电位升之和, 即  $\sum U = 0$ 。

### 1) PROTEUS ISIS 电路设计

(1) 从 PROTEUS 库中选取元器件, 如表 6-4 所示。

表 6-4 元器件明细表

元器件名称	类 别	子 类	标 识	值
BATTERY/V	ACTIVE	SOURCES	B1	12
RES/kΩ	DEVICE	GENERIC	R1、R2、R3	10、15、20

(2) 根据图 6-6 所示连接电路，并将电压表接入电路中（注意电压表的方向性）。

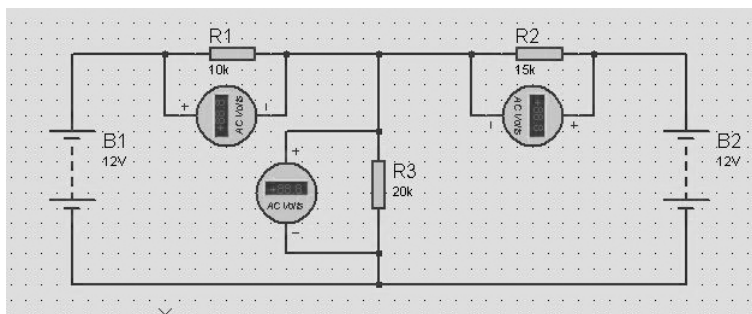


图 6-6 基尔霍夫电压定律连线图

## 2) PROTEUS ISIS 调试与仿真

(1) 如图 6-7 所示，将电路接好并接入电压表，单击仿真按钮  开启仿真。

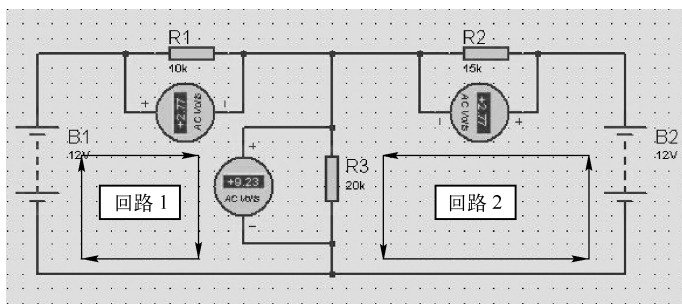


图 6-7 基尔霍夫电压定律调试与仿真

(2) 记录仿真时电路中两个回路上电压表的数值，并记入表 6-5 和表 6-6 中。

表 6-5 回路 1 电压变化

元 件	B1	R1	R3
压降/V	-12	+2.77	+9.23

表 6-6 回路 2 电压变化

元 件	B2	R2	R3
压降/V	-12	+2.77	+9.23

由表 6-5 和表 6-6 可以计算出两回路的电压降与电压升之和  $\sum U$ ，即

$$\sum U_1 = -12 + 2.77 + 9.23 = 0$$

$$\sum U_2 = -12 + 2.77 + 9.23 = 0$$

由此验证在某一闭合回路中电压的代数和等于零，即  $\sum U = 0$ 。



## 6.3 基于 PROTEUS ISIS 的戴维南定理的设计与仿真

**戴维南定理：**任何一个有源二端线性网络都可以用一个电动势为  $E$  的理想电压源和内阻  $R_0$  串联的电源来等效代替。等效电源的电动势  $E$  就是有源二端网络的开路电压  $U_0$ ，即将负载断开后  $a$ 、 $b$  两端之间的电压，等效电源的内阻  $R_0$  等于所有电源均除去后得到的无缘网络  $a$ 、 $b$  两端之间的等效电阻。

### 1. PROTEUS ISIS 电路设计

(1) 从 PROTEUS 库中选取元器件，如表 6-7 所示。

表 6-7 元器件明细表

元器件名称	类别	子类	标识	值
BATTERY/V	ACTIVE	SOURCES	B1、B2、B3	140、90、100
RES/ $\Omega$	DEVICE	GENERIC	R1、R2、R3、R4、R5	20、5、6、4、6

(2) 根据图 6-8 和图 6-9 所示连接电路，将电源 B1、B2、B3 的属性 Internal Resistance 设置为 0（默认为 0.1）。

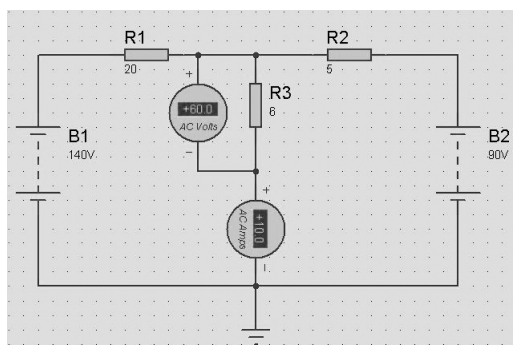


图 6-8 戴维南定理仿真电路 A

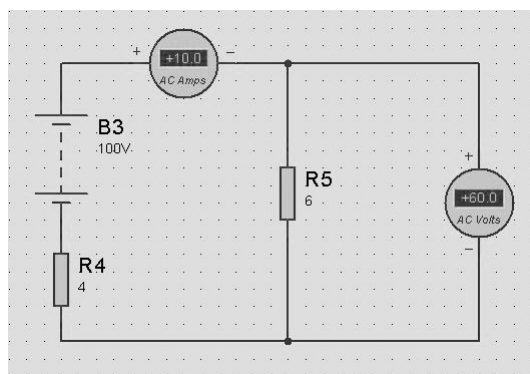


图 6-9 戴维南定理仿真电路 B

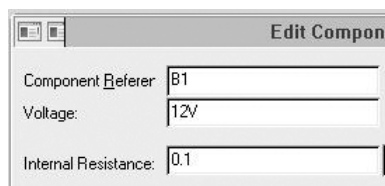


图 6-10 电源属性

### 2. PROTEUS ISIS 调试与仿真

单击仿真按钮进行仿真，分别将两电路电压表、电流表的读数记入表 6-8 中。



表 6-8 测量数值

	电压表/V	电流表/A
电路 A	100	10
电路 B	100	10

根据表 6-8 进行比较,可以看出 A 电路与 B 电路中 R3 与 R5 的电路参数完全一致,即  $U_3 = U_5$ 、 $I_3 = I_5$ , 所以对于电阻 R3 与 R5 来说两个电路是等效的,即电路 A 中电源 B1、B2 和电阻 R1、R2 所组成的有源二端网络等效于理想电压源 B3 与电阻 R4 串联的电源。即

$$E_{B3} = E_{B1} - R_1 \times I_1 = 100V$$

$$R_4 = R_1 \times R_2 / (R_1 + R_2) = 4\Omega$$

由此可以验证戴维南定理的正确性。



## 6.4 基于 PROTEUS ISIS 的诺顿定理的设计与仿真

任何一个有源二端线性网络都可以用一个电流为  $I_s$  的理想电流源和内阻  $R_0$  并联的电源来代替,等效电源的电流  $I_s$  就是有源二端网络的短路电流,即将  $a$ 、 $b$  两端短接后其中的电流,等效电源的内阻  $R_0$  等于有源二端网络中所有电源除去(理想电压源短路,理想电流源开路)后所得到的无源网络  $a$ 、 $b$  两端之间的等效电阻。这就是诺顿定理(如图 6-11 所示)。

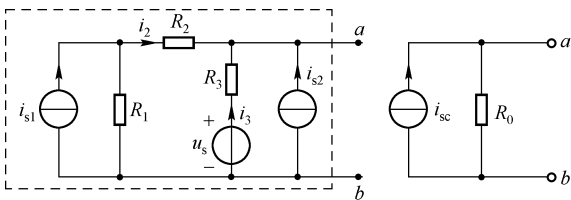


图 6-11 诺顿定理

### 1. PROTEUS ISIS 电路设计

(1) 从 PROTEUS 元件库中选取元器件,如表 6-9 所示。

表 6-9 元器件明细表

元器件名称	类 别	子 类	标 识	值
BATTERY/V	ACTIVE	SOURCES	B1、B2、B3	140、90、100
RES/ $\Omega$	DEVICE	GENERIC	R1、R2、R3、R0	20、5、6、4

(2) 完成电路设计,如图 6-12 和图 6-13 所示。

### 2. PROTEUS ISIS 调试与仿真

单击仿真按钮进行仿真,观察两电路的电流表读数并记入表 6-10 中。

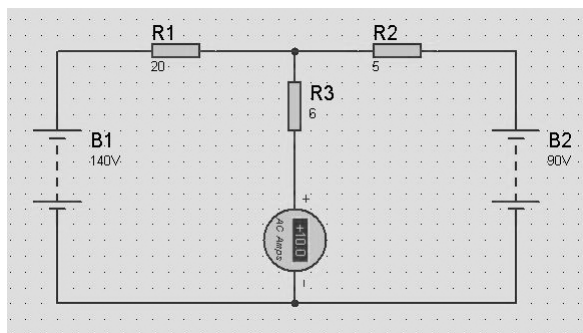


图 6-12 诺顿定理原理图 A

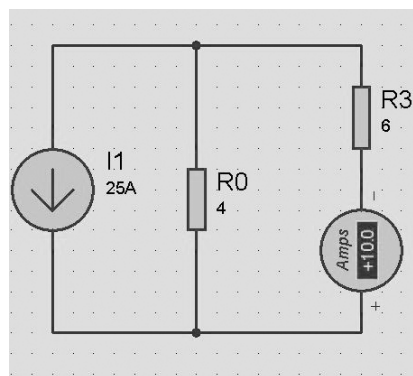


图 6-13 诺顿定理原理图 B

表 6-10 电流测量值

	电流值/A
电路 A	10
电路 B	10

根据表 6-10 进行比较,可以看出两电路中流过  $R_3$  的电流是完全一样的,所以对于  $R_3$  来说两电路是等效的,即电路 A 中电源  $B_1$ 、 $B_2$  和电阻  $R_1$ 、 $R_2$  所组成的有源二端网络等效于理想电流源  $I_1$  与电阻  $R_0$  并联的电源。



## 6.5 基于 PROTEUS ISIS 的三相交流电路的设计与仿真

三相电源一般来自发电机或变压器二次侧的三个绕组。三个绕组的始端为 A、B、C,若将三个绕组的末端连在一起,便形成星形连接。三个绕组的连接点成为一个公共端,称为地线,从三个绕组始端引出的三条输电线称为端线或相线(俗称火线)。这时电源有四条输电线,称为三相四线制电源。一般三相电源能提供三个具有相同频率和振幅但在相位上彼此相差  $120^\circ$  的正弦电压,这组电压叫做对称三相电压。

### 1. PROTEUS ISIS 电路设计

1) 从 PROTEUS 元件库中提取元器件 仿真实验中,利用三个频率为 50Hz、有效值为 220V、相位各差  $120^\circ$  的正弦信号源代替三相交流电,每相均采用 220V、100W、内阻为  $484\Omega$  的灯泡作为负载。线路采用星形三相四线制连接,即三相交流电源的公共端 N 与三相负载的公共端相连。

仿真电路所使用的元器件见表 6-11。

表 6-11 元器件明细表

元器件名称	所属类	所属子类	标识	值
LAMP	ACTIVE	LAMP	L	—
VSINE	VSINE	ASIMMDLS	V	—

2) 电路设计 放置元器件, 放置电源和地, 连接线路, 进行元器件属性设置、电气检测, 所有操作都是在 ISIS 中进行的。

## 2. PROTEUS ISIS 调试与仿真

(1) 按图 6-14 所示连接好实训电路。

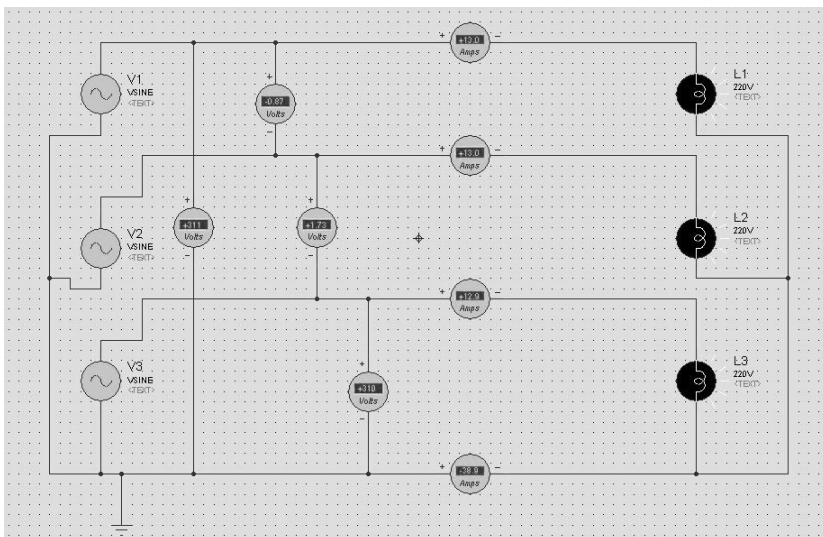


图 6-14 三相交流电路原理图

(2) 设置交流信号的参数, 见表 6-12。

表 6-12 交流信号参数

元件参考	幅值/V	频率/Hz	时间延迟/ms
V1	311	50	0
V2	311	50	6 ~ 67
V3	311	50	-6 ~ 67

(3) 单击仿真按钮, 启动仿真。

(4) 测量对称负载, 各项参数测量得数略。

(5) 测量不对称负载 (A 相负载中并联一个白炽灯泡), 各项参数测量得数略。



## 6.6 思考与练习

1. 设计电路验证基尔霍夫定律。
2. 设计电路验证戴维南定理。
3. 设计电路验证诺顿定理。
4. 设计测试电路仿真电容充、放电过程。
5. 设计电路仿真正弦交流电路功率的测试。

# 第7章 模拟电子电路的仿真与设计

在电子线路中,研究含有线性电阻、电容和电感元件的电路(有直流和交流稳态及动态电路),属于基础理论电路,相对比较容易理解;而研究以非线性元件二极管和三极管为核心的电路,因为交直流并存,元件的多种作用与工作电压的变化密切相关,不便于初学者理解和掌握。模拟电路就是以三极管为核心,处理模拟信号的电路。初学者往往对模拟电子技术的学习感到困惑,概念多而抽象,学生在实验室里做实验的时间又有限,尤其是前几章的内容,稀里糊涂就过去了,给后面的学习带来不少困难。在这一章里,我们主要对模拟电路中的几个典型实验进行仿真和分析,然后给出两个模拟电路设计的综合实例,目的在于使读者对这一部分的仿真元件及仪器有个基本的掌握,帮助大家更好地理解和学习模拟电子技术,借助软件,反复学习和验证,最终克服学习中的难关。



## 7.1 模拟电路常用器件与仪器

### 1. 模拟电路常用器件

模拟电路中常用的器件主要有三极管、二极管、电阻、电容、电感、变压器、直流电源、信号源、集成运放等。下面我们来看这些器件如何拾取。

1) 三极管 如何在 PROTEUS 的浩瀚器件库中找到自己想要的三极管器件呢? 打开 PROTEUS 的器件拾取对话框,在类别“Category”中的“Transistors”子类就是三极管,单击“Transistors”,出现如图 7-1 所示的元件。这些器件和我们平时常用的国产三极管的型号不太一致,比如常用的国产高频小功率管 3DG6 对应于 2N5551,替换的原则是双方的管型一致,另外参数也要一样(当然根据设计需求允许有误差),器件替换对应也可以在网上查找。如果只是一般的原理仿真,可以直接输入“NPN”或“PNP”来拾取通用器件。如果用到场效应管,则可以在对应的子类中查找,如图 7-1 中右侧所示。

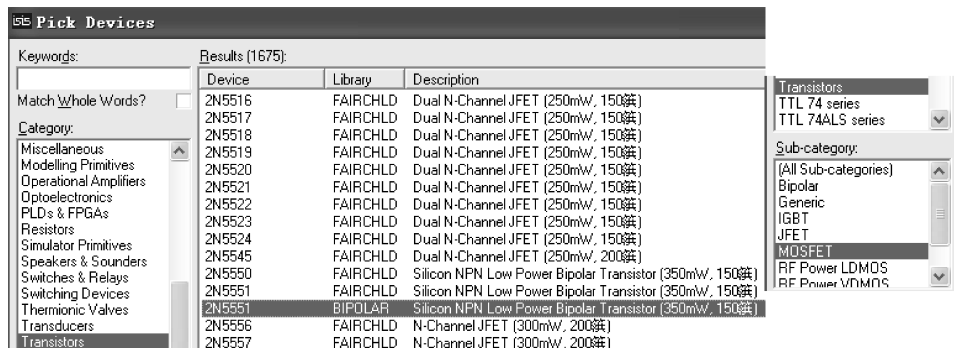


图 7-1 三极管器件拾取对话框

2) 二极管 二极管的种类很多,包括整流桥、整流二极管、肖特基二极管、开关二极管、隧道二极管、变容二极管和稳压二极管。打开 PROTEUS 的器件拾取对话框,选择“Category”中的“Diodes”,出现如图 7-2 所示的对话框。一般来说,选取子类“Sub-category”中的“Generic”通用器件即可,图 7-2 右边给出了通用器件的查寻结果,可以单击看看需要使用哪种器件。

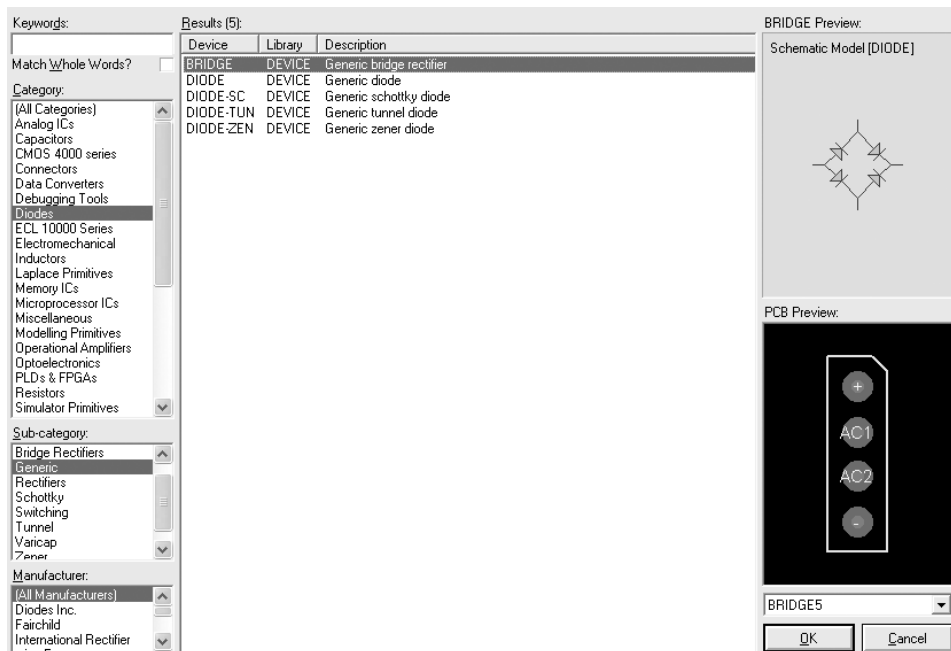


图 7-2 二极管器件拾取对话框

3) 电阻 电阻的分类为“Resistors”,子类有 0.6W 和 2W 金属膜电阻,3W、7W 和 10W 线绕电阻,通用电阻,热电阻 (NTC),排阻 (Resistor Packs),可变电阻 (Variables) 及家用高压系列加热电阻丝。

常用电阻可直接输入通用电阻“RES”拾取,然后再修改参数。这里我们主要说一下比较常用的可变电阻,直接输入“POT”或“POT-”可找到四个或三个相关器件。

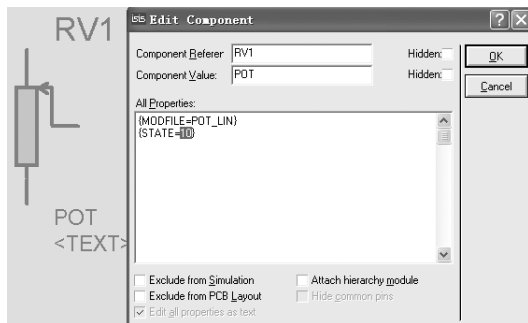


图 7-3 滑动变阻器元件属性对话框

“POT”为一般滑动变阻器,触头不能拉动,需选中后打开元件属性对话框,修改“STATE”来改变触头的位置。“STATE”的初始值为 5,触头位于中间,改为 10 后,触头位于最上面,如图 7-3 所示。由于调整不方便,一般不使用此元件,而使用下面的几个滑动变阻器。

“POT-HG”滑动变阻器的好处是可以直接用鼠标来改变触头位置,精确度和调整的最小单位为阻值的 1%,比如一个 1kΩ 的电阻,精确到 10Ω,而一个 100kΩ 的电阻只能精确到 1kΩ。所以,当电阻较大时,考虑把它分成两部分串联,一部分为较大阻值的固定电阻,另一部分为较小阻值的滑动电阻,这样比较科学。

“POT-LIN”和“POT-LOG”滑动变阻器与“POT-HG”一样可以通过鼠标来改变触头位置，但精确度和调整的最小单位均为阻值的10%。

**4) 电容** 模拟电路中常用的电容为极性电容，即电解电容。其实无极性电容和电解电容在使用时没有什么区别，只不过当电容值较大，通常在 $1\mu\text{F}$ 以上时，要做成电解电容。放大电路中的耦合电容一般为 $10 \sim 100\mu\text{F}$ ，为电解电容。特别要注意的是，电解电容正极性端的直流电位一定要高于负极性端才能正常工作，否则会出现意外事故。

常用的无极性电容的名称为“CAP”，极性电容为“CAP-ELEC”，还有一个可动画演示充放电电荷的电容为“CAPACITOR”。极性电容“CAP-ELEC”的原理图符号正端不带填充，负端方框中填充有斜纹。使用时可直接输入名字拾取。

**5) 电感和变压器** 电感和变压器同属电感“Inductors”这一分类，只不过在子类中又分为通用电感、表面安装技术（SMT）电感和变压器。一般来说，使用电感时直接拾取“INDUCTOR”元件，使用变压器时，要根据原、副边的抽头数而定。

打开元件拾取对话框，选取“Inductors”大类下的子类“Transformers”，如图7-4所示，在右侧显示出变压器可选元件。常用的是前4种，名称前缀为“TRAN-”，也可以直接输入这个前缀来搜寻变压器。为了帮助大家记忆变压器的名称，以第一个变压器“TRAN-1P2S”为例来说明它的含义。“TRAN”是变压器的英文“TRANSFORMER”的缩写，“P”是原边“PRIMARY”的意思，“S”是副边“SECONDORY”的意思。而后面三个变压器都是饱和变压器，如“TRSAT2P2S2B”即Saturated Transformer with secondary and bias windings，意思是具有副边和偏置线圈的饱和变压器。

Sub-category:	Results (7):		
(All Sub-categories)	Device	Library	Description
Generic	TRAN-1P2S	DEVICE	Single primary, two insulated secondaries
SMT Inductors	TRAN-2P2S	DEVICE	Simple transformer
Transformers	TRAN-2P3S	DEVICE	Transformer with centre tapped secondary winding
	TRAN-2P5S	DEVICE	Transformer with tri-tapped secondary winding
	TRSAT2P2S	DEVICE	Saturated Transformer
	TRSAT2P2S2B	DEVICE	Saturated Transformer with secondary and bias windings
	TRSAT2P3S	DEVICE	Saturated Transformer with centre tapped secondary

图 7-4 变压器拾取对话框

变压器在调用时，由于对称按钮可能处于选中状态，原、副边绕组的位置就颠倒了，使用时要注意，尤其是原边和副边绕组数目相同的变压器，这涉及原、副边的匝数比是升压或降压变压器的问题。

变压器的匝数比是通过改变原、副边的电感值来实现的。打开“TRAN-2P2S”变压器的元件属性对话框，如图7-5所示，原边和副边的电感值都是1H，即变比 $n$ 为1:1。如果想使它成为 $n=10:1$ 的降压变压器，可以改变原边电感，也可以改变副边电感，还可以两者同时改变，但要保证原、副边电压比值等于原边电感与副边电感的平方比。

改变原、副边的电感值分别为100H和1H（也可以为1H和0.01H），即原、副边电压比为10:1，此变压器为降压变压器，如图7-6所示。

变压器变比设定后，在原边加一个交流源“ALTERNATOR”，使它的幅值为100V，频率为50Hz，同时也在原边加一个交流电压表，在副边也加一个交流电压表，运行仿真，显示原边电压有效值为70.7V，副边电压有效值为7.07V，变压比为10:1，如图7-7所示。

**6) 交、直流电源** 通常有单电池“CELL”和电池组“BATTERY”两种，可任意改变其值。单相交流电源为“ALTERNATOR”，可改变其幅值（半波峰值）和频率，如图7-8所示。

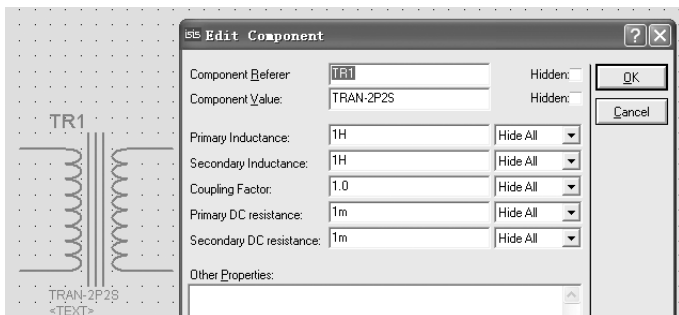


图 7-5 变压器属性对话框

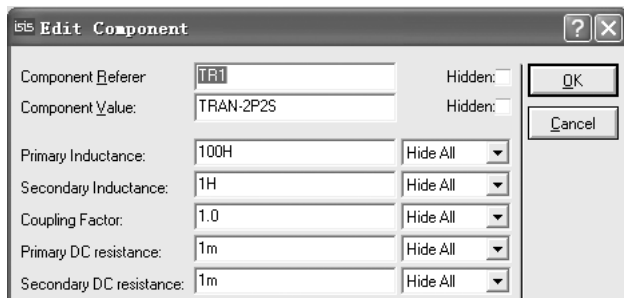


图 7-6 修改变压器变比

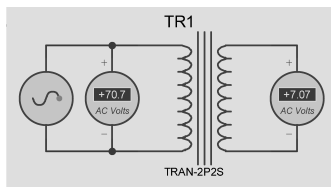


图 7-7 变压器变比仿真

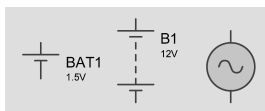


图 7-8 交、直流电源

7) 集成运放 打开元件拾取对话框, 选取“Operational Amplifiers”分类, 显示子类有“Dual”、“Deal”、“Octal”、“Quad”、“Single”、“Triple”, 分别为双运放 (即一个集成芯片内所包含的两个相同运放)、理想运放、八运放、四运放、单运放和三运放。常用的集成运放是通用的理想运算放大器, 可直接选子类“Deal”中的“OP1P”。如果知道集成运放的名称, 也可以直接查寻, 比如对常用的四运放 LM327 直接输入“LM327”即可。

## 2. 模拟电路仿真中的常用仪器

模拟电路中常用的仿真仪器主要有交流电压表、交流电流表、直流电压表、直流电流表、信号发生器、示波器和扬声器。

单击工具栏中的虚拟仪器图标, 如图 7-9 所示, 在对象选择区出现所有的虚拟仪器名称列表, 其中“OSCILLOSCOPE”、“SIGNAL GENERATOR”、“DC VOLTMETER”、“DC AMMETER”、“AC VOLTMETER”、“AC AMMETER”分别为示波器、信号发生器、直流电压表、直流电流表、交流电压表和交流电流表。

交、直流电压表和交、直流电流表的量程都可以设定, 比如可以设定一个交流电压表为

毫伏表,如图7-9所示,只需改变元件属性中的“Display Range”为“Millivolts”即可。

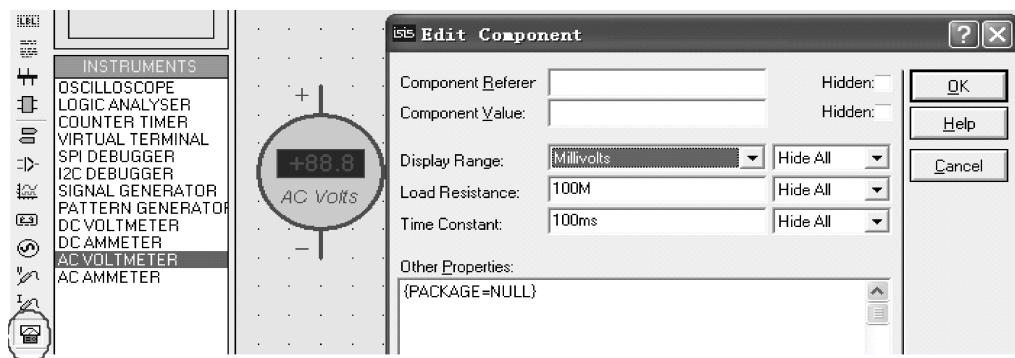


图 7-9 交流毫伏表的量程设定

信号发生器的用法在前面章节已经介绍,下面主要说明示波器和扬声器的用法。

1) 示波器 PROTEUS 的虚拟示波器能完成4个通道(A、B、C、D)的波形显示与测量。待测的4个输入信号分别与示波器的4个通道连接,信号的另一端应接地。

在前面的例子中我们已经熟悉了示波器的原理图符号,下面介绍仿真运行后示波器界面各部分旋钮的功能。图7-10是示波器仿真运行后的界面。

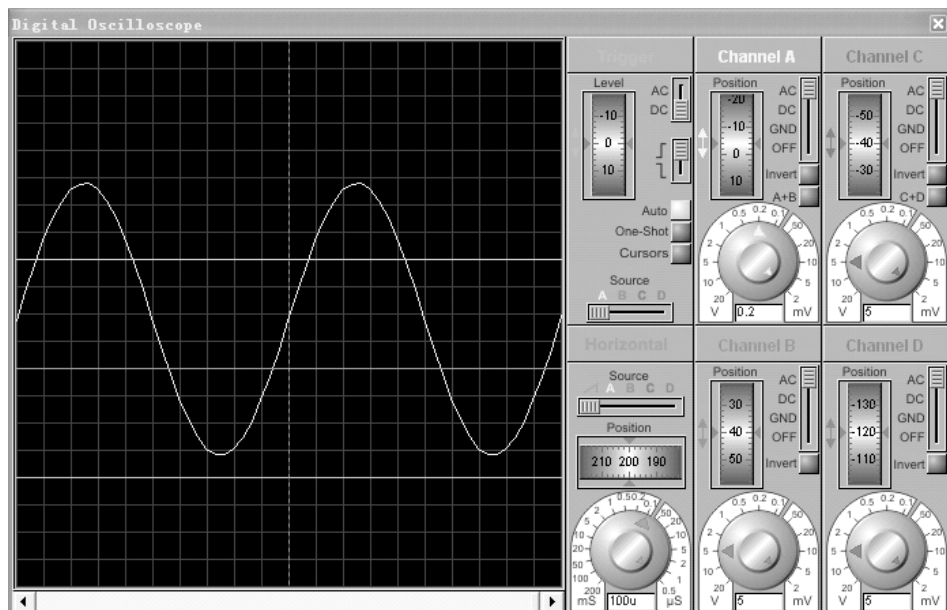


图 7-10 示波器仿真运行后的界面

以通道A为例,“Position”旋钮用来调整波形的垂直位移,下面的旋钮用来调整波形的幅度显示比例,外面的黄色箭头是粗调,里面的黄色小箭头是细调,当读刻度时,应把里层的箭头顺时针调到最右端。4个通道的对应旋钮使用方法一样。在“Horizontal”下方的两个旋钮分别用来调整波形的水平位移和扫描频率。当用鼠标单击黑色的波形显示区域后,也可以通过滚动鼠标滑轮来调整扫描频率。其他旋钮可保持原位不动。

在运行过程中如果关闭示波器,需要从主菜单【Debug】中选取最下面的【VSM Oscil-



loscope】来重新打开。

2) **扬声器** 扬声器在模拟电路的仿真中也经常用到。可直接输入“Speaker”来调用，两个接线端不分正负，因为它接收的是交流模拟信号。要注意驱动信号的幅值和频率应在扬声器的工作电压和频率范围之内，否则不会响。当扬声器不鸣响时，可能是因为信号种类不匹配（比如数字信号）或扬声器的电压设得太大而需要修改。扬声器的属性参数对话框如图 7-11 所示。

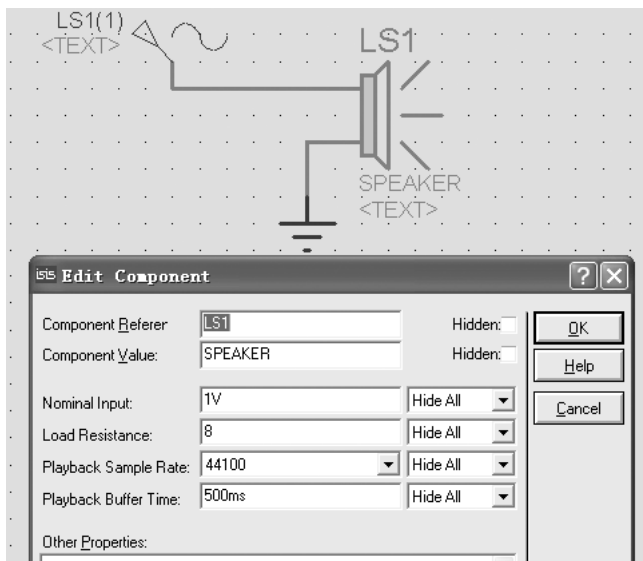


图 7-11 扬声器属性参数对话框



## 7.2 单管共射放大器及负反馈

在模拟电路中，单管共射放大器及负反馈是非常重要的内容，这两个实验通常合并在一块电路板中来做。接下来我们讨论单管共射放大器的静态工作点及动态参数的调试与测量，最后再观察两级放大电路中负反馈带来的影响。

### 1. 单管共射放大电路静态工作点的调试

单管共射放大器及负反馈实验的电路如图 7-12 所示。我们先考虑单管共射放大器部分，即前一级电路，如图 7-13 所示。

照图 7-12 把整个电路图连接好，两级参数可取相同。接上直流电源、信号发生器和示波器。下面调试第一级的静态工作点，即找到一个合适的静态工作点，然后再用直流表测量出来。

把开关按图示位置设定好，按仿真运行按钮，将信号发生器的频率调为 1kHz，幅值尽可能大，直到观察到示波器显示的输出波形出现双顶失真为止，如图 7-14 中的波形（a）所示。看看这个失真的波形是否上下对称失真，如果不对称，调整图 7-13 中的滑动变阻器 RV1 来改变静态工作点使波形看似对称，如图 7-14 中的波形（b）所示。因为眼睛看到的

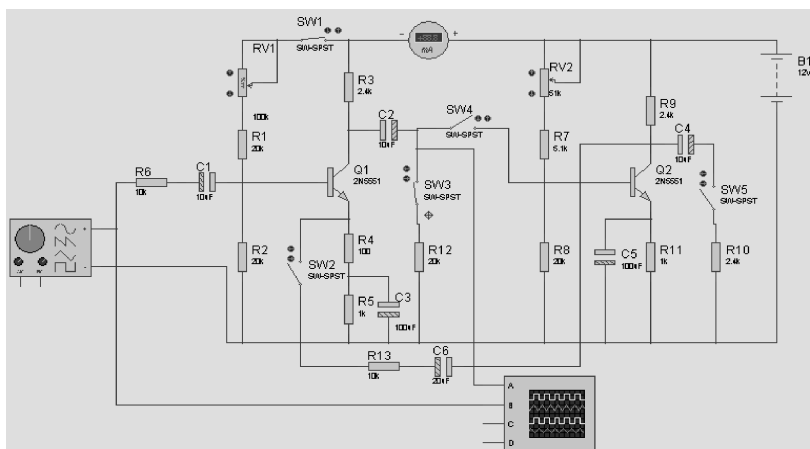


图 7-12 单管共射放大器及负反馈实验电路图

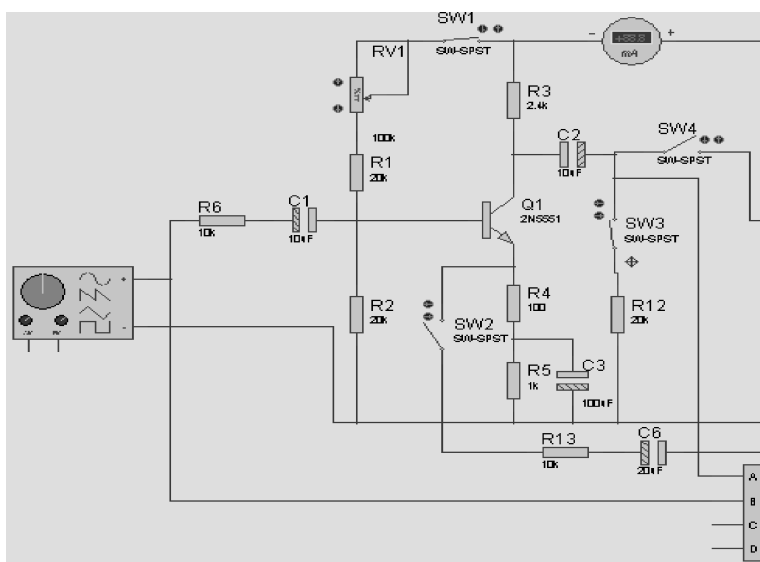


图 7-13 单管共射放大器实验电路图

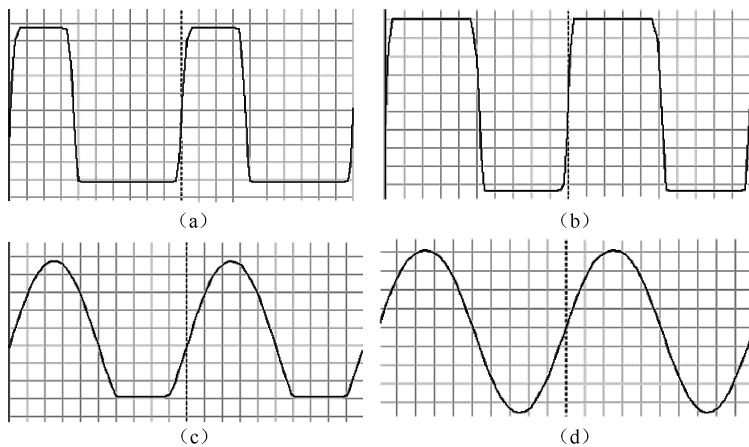


图 7-14 单管共射放大器调试静态工作点波形

对称失真并不一定是真的对称, 所以还需减小信号发生器的幅值, 使波形一端的失真刚好消失, 如图 7-14 中的波形 (c) 所示, 这验证了静态工作点仍然不合适。进一步调整滑动变阻器, 使波形两端出现对称失真, 再减小信号发生器的幅值, 使波形一顶失真消失, 反复几次, 直到波形两顶的失真刚好同时消失, 如图 7-14 中的波形 (d) 所示, 这时的静态工作点

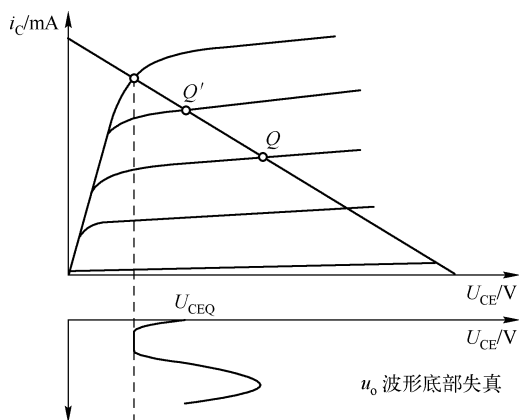


图 7-15 三极管的输出特性与静态工作点

上述静态工作点的调整方法, 就是故意让输出波形失真来看失真的对称度, 从而判断静态工作点是否位于交流负载线的中间, 因为合适的静态工作点并不意味着不会产生失真, 只要输入信号足够大, 就会产生失真, 只不过是产生对称的失真。通过反复调整输入信号的幅值和基极电阻的大小, 来观察和改变静态工作点, 从而找到一个最佳静态工作点, 只有找到了最佳静态工作点, 接下来的动态参数测量才有意义。给定一块电路板, 不能盲目地去进行数据测量。

虽然电容隔直, R6 左边交流信号源的短路线可以省去, 开路即可, 但在没有电容的直接耦合电路中却不能开路, 为了养成良好的习惯, 建议使交流信号短接而不是开路。如图 7-16 所示, 把三个直流电压表和一个直流电流表 (毫安表) 连接起来, 可测得如表 7-1 所示的数据。其中,  $I_C$  的数据是约等于。

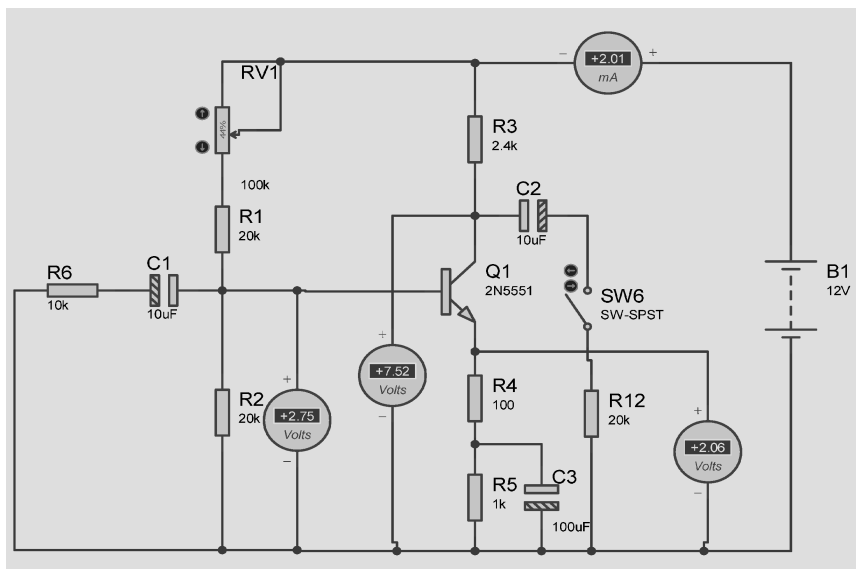


图 7-16 静态工作点的测量

表 7-1 单管共射放大电路静态工作点测量值

$V_B/V$	$V_E/V$	$V_C/V$	$I_C/mA$	$V_{BE}/V$	$V_{CE}/V$
2.75	2.06	7.52	2.01	0.69	5.46

注意：三个电压表一定要直接连接到三极管的三个极上，不能在电容 C1 前或电容 C2 后测量。表 7-1 中后两列是计算值。

2. 单管共射放大电路动态参数的测量

前面提到，静态工作点的合适与否直接影响交流输出信号的幅值。那么是不是有了合适的静态工作点后，输出电压与信号源的比就一定能得到最大值呢？不是的，影响放大倍数的还有放大电路的几个动态参数：输入电阻、输出电阻和带宽。首先来讨论电压放大倍数的测量。

1) 电压放大倍数 电压放大倍数有两种含义，一种是输出电压对信号源的比值，另一种是输出电压对输入电压的比值。由于 PROTEUS 的虚拟信号源都是理想电压源，没有内阻，所以图 7-16 所示的电路中用电阻 R6（10kΩ）来模拟信号源内阻，当然与实际相比有些大了。真正分到放大电路中的信号电压是 R6 的右端（即基极）到地之间的交流电压，另外一部分电压降落在 R6 上。

在输入端接信号发生器，在信号发生器上并联一个交流毫伏表以测量信号源电压的有效值。调节信号发生器的幅值使交流毫伏表的读数约为 10mV，把示波器接在输出端，观察输出波形，以不失真为准。断开负载电阻使放大电路空载，在输出端接交流电压表。运行仿真，各表读数如图 7-17 所示。可计算：

$$A_{vs} = \frac{V_o}{V_s} = \frac{1\ 380}{10} = 138$$
$$A_v = \frac{V_o}{V_i} = \frac{1\ 380}{5.07} \approx 242$$

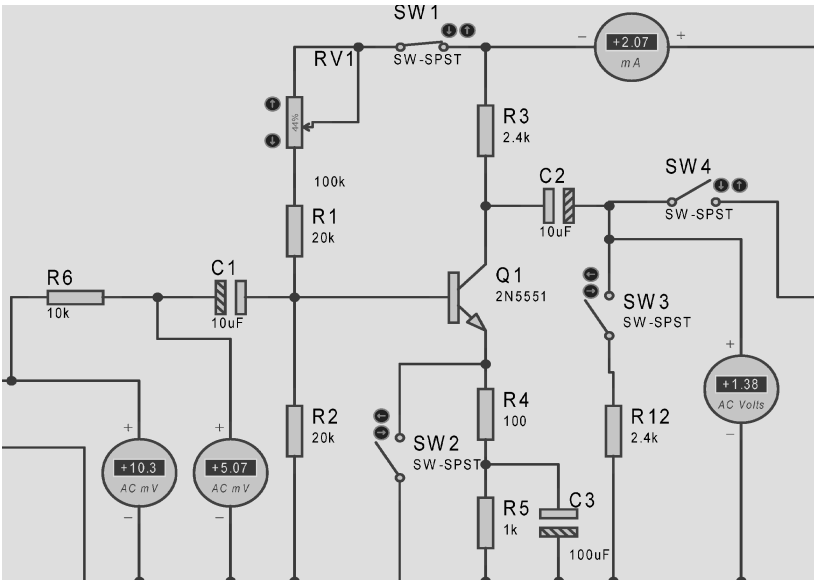


图 7-17 空载时电压放大倍数的测量

合上开关 SW3 加上负载后, 输出电压的幅值会下降, 请大家自己测量和计算并分析原因。

由于是单管共射放大电路, 所以输出波形和输入波形是倒相的, 放大倍数应该是负值。示波器的输入、输出波形如图 7-18 所示。

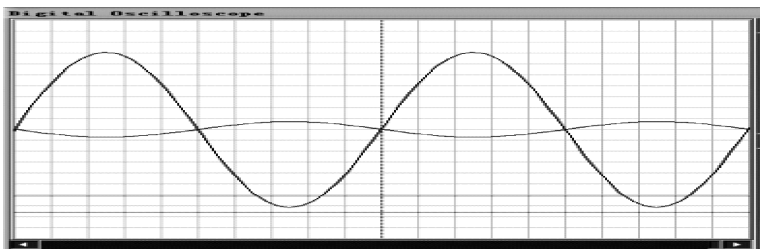


图 7-18 输入与输出波形

**2) 输入电阻** 放大电路的输入电阻是从放大电路输入端看进去的无源网络的等效电阻, 计算此电阻要先画出放大电路的微变等效电路, 也可以直接通过实验方法来测量, 这样更方便和快捷。其原理如下。

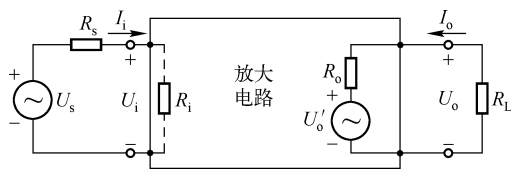


图 7-19 输入、输出电阻测量原理图

如图 7-19 所示, 可以把放大电路的交流通路看成二端口网络, 输入端为含有内阻的信号源, 输出端接负载。其中,  $R_i$  为输入电阻,  $R_o$  为输出电阻。当电路设计好后, 二端口的参数就确定不变了。空载时, 输出电压  $U_o$  与输入电压  $U_i$  的比值是定值, 但由于二端口外的元件  $U_s$ 、 $R_s$  及  $R_i$  是随不同的用户使用而定的, 所以根据两端串联分压原理,  $R_i$  与  $R_o$  会分别影响  $U_i$  与  $U_o$  的值, 从而引起输出电压的变化而影响电压放大倍数。

在图 7-19 中的输入端, 如果知道  $U_s$ 、 $R_s$  及  $U_i$ , 就可以算出  $R_i$ , 测量输入电阻的原理就是如此。

回头看实验图 7-17,  $U_s = 10.3\text{mV}$ ,  $U_i = 7.07\text{mV}$ ,  $R_s = 10\text{k}\Omega$ , 则可算出  $R_i = 10.7\text{k}\Omega$ 。可见, 输入电阻越大, 放大电路分得的电压越大, 输出电压就越大, 当然这里模拟用的内阻  $R_s$  有点过大。

**3) 输出电阻** 根据图 7-19 不难理解, 如果把放大电路看成一个电压源, 对负载供电, 则输出电阻  $R_o$  就是这个电压源的内阻,  $R_o$  越小, 负载上分得的电压越大, 放大电路的性能就越好。因此有  $U_o = \frac{R_L}{R_L + R_o} U'_o$ , 其中  $U'_o$  为空载电压。

测量输出电阻的实验是, 分别测出空载和带负载情况下的输出电压  $U_o$  及负载电阻  $R_L$ , 就可以算出  $R_o$  的值。

**4) 带宽** 前面分别提到静态工作点、交流输入及输出电阻会影响放大电路的电压放大倍数, 但当这些参数都设计合理后, 是不是放大电路的性能就完美无缺了呢? 其实, 前面提到的放大都是对某一固定频率信号的幅值进行的放大, 我们在做实验的时候, 通常把信号频率调节到  $1\text{kHz}$ 。如果保持信号的幅值不变而改变其频率, 会发现放大倍数在某些频段会保持不变, 而在另一些频段则会突然下降, 甚至为零。这就是我们所说的频率响应, 即频率对放大倍数的影响。不同放大电路的频率响应是不一样的, 这主要是因为电路中电容 (耦合电容、

旁路电容、极间电容等)的阻抗会随频率而改变,从而导致电路的输入、输出阻抗变化,影响输出电压的大小。在单管共射放大电路中,频率过高和过低都会造成放大倍数的衰减,只有在中频段,放大倍数才稳定不变,这一段的频率范围称为带宽,通常用 $f_{BW}$ 来表示。

测量单管共射放大电路带宽的方法是,在放大电路输入端先加一个小信号,比如 10mV、1kHz 信号,用示波器观察输出电压波形,要保证输出波形不失真。调节示波器的扫描旋钮,让波形集中,调整示波器的垂直增益,使输出波形正好占据 10 格,如图 7-20 所示。

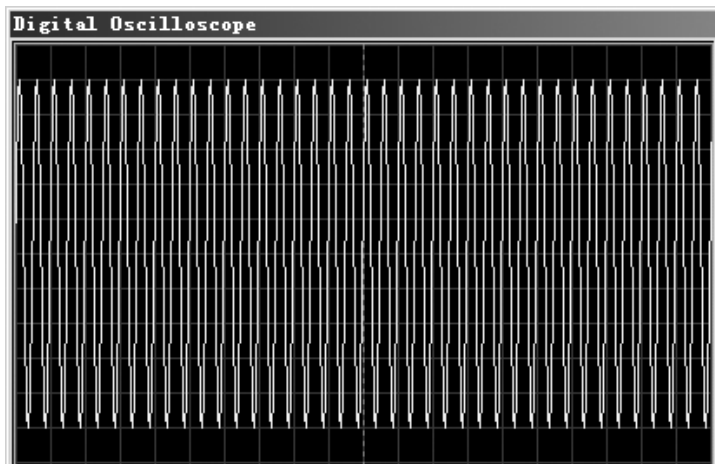


图 7-20 中频段输出波形的幅度

接下来减小信号发生器的频率,调整示波器的扫描旋钮,使波形在频率较低的情形下仍能相对集中,以便观察幅值所占的格数。继续减小信号发生器的频率值,直到输出波形在示波器中所占的格数减为 7 格,如图 7-21 所示,这时读信号发生器的频率为 13Hz,即放大电路的下限转折频率 $f_L$ 。以同样的方法读出上限转折频率为 700kHz,即 $f_H$ 。这个放大电路的带宽 $f_{WH} = f_H - f_L$ ,约为 700kHz。

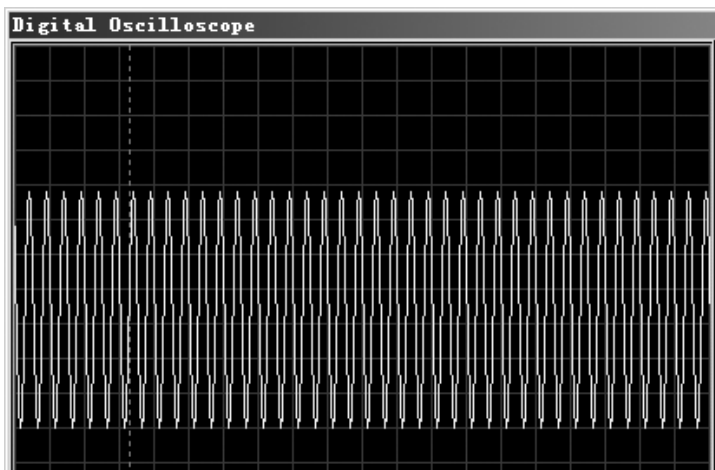


图 7-21 截止频率时输出波形的幅度

根据单管共射放大倍数频率响应的幅频特性,如图 7-22 所示,在中频段,即 13Hz ~ 700kHz 范围内,放大倍数基本恒定,当频率小于 13Hz 或大于 700kHz 时,放大倍数按每 10 倍

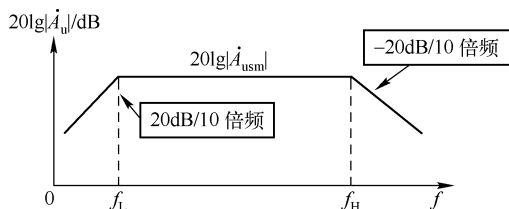


图 7-22 单管共射放大电路的频率响应

频段 20 分贝的速度上升或衰减。图中显示的是理想幅频特性，其实，在转折频率处，中频放大倍数要下降 3dB，即是原来的 0.707 倍。

在本实验中，因为输入信号未变，输出电压变为原来的 0.7 倍，即放大倍数变为原来的 0.7 倍。故改变信号频率，使输出电压的幅值由原来的 10 格缩为 7 格时，即转折频率所对

应的幅值，就测出了转折频率和带宽。

### 3. 负反馈

在单管共射放大电路实验的一开始，我们给出的实验电路板是一个含有负反馈的两级放大电路，如图 7-12 所示。可以判断出该负反馈为电压串联负反馈。因为电压串联负反馈能够使放大电路的输出电阻变小，输入电阻增大，拓展了带宽，从而改善了该两级放大电路的动态性能，但是所有这些都是以牺牲放大倍数为代价的。

本实验我们先分别调试好两级放大电路的静态工作点，然后在断开负反馈的情况下，按照前面介绍的方法，测试两级放大电路的整体输入电阻、输出电阻、电压放大倍数和带宽。合上开关 SW2，在接上负反馈的情况下，把以上 4 个参数再测量一遍，进行比较，找出它们之间的关系。

为了节约时间，4 个参数的测量应该这样来做比较方便，比如测输入电阻，一次把含有负反馈和不含负反馈两种情况都测了。先把开关 SW2 拨到断开位，读输入端信号电压，然后再把开关 SW2 拨到闭合位，重读一遍。其他参数依次类推。

通过测量和比较，得出的结果是：

$$A_f = \frac{A}{1 + AF}$$

$$R_{if} = (1 + AF)R_i$$

$$R_{of} = \frac{1}{1 + AF}R_o$$

$$f_{BWF} = (1 + AF)f_{BW}$$



## 7.3 射极跟随器

共集电极放大电路又叫射极跟随器，因为它的电压从射极输出且与输入电压大小几乎相等，相位一致，就好像输出电压总是跟随输入电压的变化一样。射极跟随器并不能放大电压，但它能放大电流，它的输入电阻高，输出电阻低，电路的动态性能比较好，适合做多级放大电路的初级和末级。但由于差动放大电路和功率放大电路的出现，在高性能运放的输入级和输出级一般不使用射极跟随器。如图 7-23 所示，是射集跟随器实验电路，在该实验中，我们主要完成以下工作：

- 测量静态工作点；
- 测量动态参数；

➤ 观察输入、输出波形。

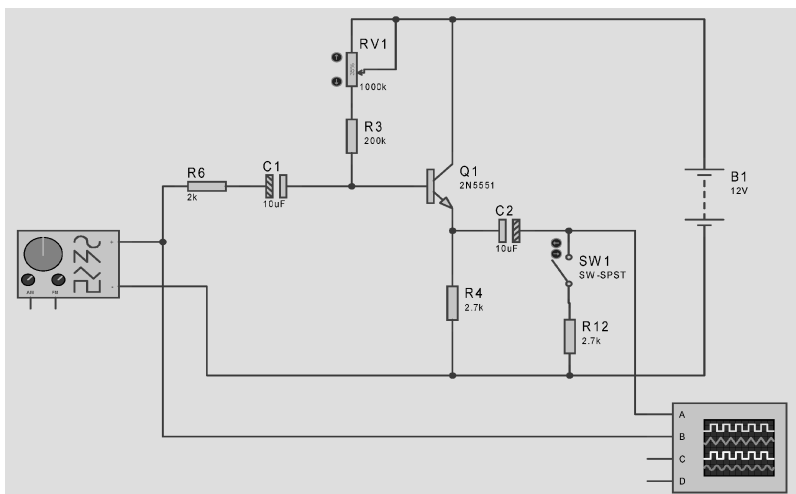


图 7-23 射极跟随器实验电路

### 1. 静态工作点的测试

按照图 7-23 的接线，先进行静态工作点调试，由于射极跟随器的电压不能被放大，所以在调试静态工作点时需要加比较高的输入电压才能观察到失真的出现，一般从 1V 加起，逐渐加大。静态工作点调整合适后，按照图 7-24 接线，测量静态工作点。注意，测量静态工作点所用的都是直流表，测量集电极电位的电压表可省去，因为集电极直接接 12V 直流电源。电流表要在属性对话框中改为毫安表，注意电流从正端流向负端。测得的数据如图 7-24 中各电表所示，列于表 7-2 中。

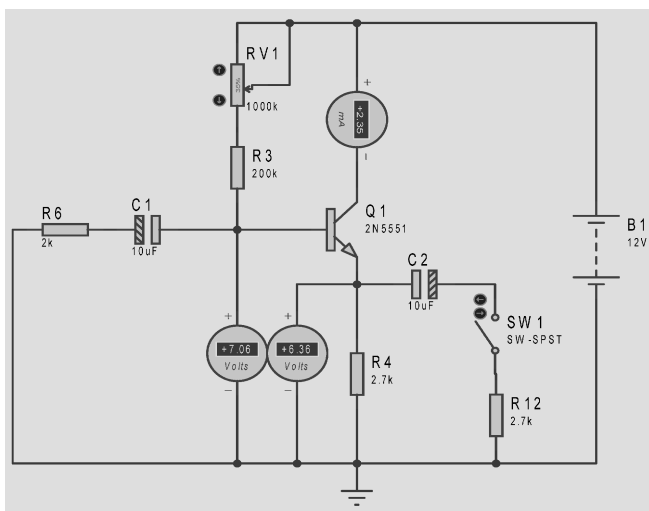


图 7-24 射极跟随器的静态工作点测量电路

表 7-2 射极跟随器静态工作点测量值

$V_B/V$	$V_E/V$	$V_C/V$	$I_C/mA$	$V_{BE}/V$	$V_{CE}/V$
7.06	6.36	12	2.35	0.7	5.64



## 2. 测量动态参数

动态参数仍然是电压放大倍数、输入电阻、输出电阻和带宽。按照图 7-23 接线, 运行仿真, 把信号发生器的频率调为 1kHz, 调节信号发生器的幅度使电路的输入电压为 2V, 测得信号发生器的电压有效值为 2.02V, 输出电压有效值为 1.99V。于是可以算出各动态参数。

### 1) 电压放大倍数

$$A_v = \frac{U_o}{U_i} = \frac{1.99}{2.00} = 0.995$$

$$A_{vs} = \frac{U_o}{U_s} = \frac{1.99}{2.02} = 0.985$$

### 2) 输入电阻

$$R_i = \frac{U_i}{U_s - U_i} R_s = \frac{2.00}{2.02 - 2.00} \times 2 = 200\text{k}\Omega$$

3) 输出电阻 保持输入信号不变, 空载和接负载时分别测得输出电压, 如图 7-25 和图 7-26 所示, 可计算输出电阻如下:

$$R_o = \frac{U'_o}{U_o} R_L = \frac{1.99 - 1.97}{1.97} \times 2.7 = 27.4\Omega$$

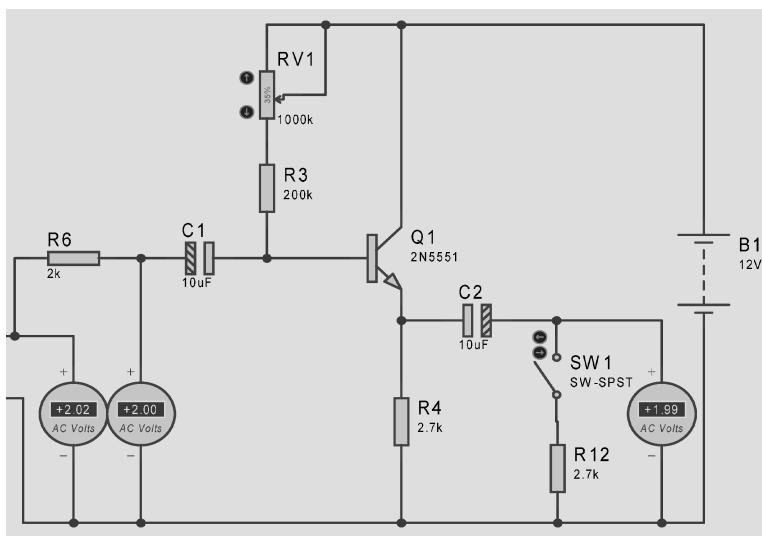


图 7-25 射极跟随器动态参数的测量

通过计算可以发现, 此射极跟随器的输入电阻高达 200kΩ, 输出电阻低至 27.7Ω, 电压放大倍数接近 1 但小于 1。至于带宽, 读者可以借鉴前面单管共射放大电路的方法自己测算。

## 3. 观察输入、输出波形

在以上各动态参数的测量过程中, 前提是输出电压不失真, 最好是把示波器接在输出端, 每测量或改变一个参数, 都要观察输出电压的波形, 确保是在不失真的状态下。

现在的主要目的是观察、对比输入、输出电压波形, 验证输出电压是否与输入电压大小相等、方向相同。在输入端加上使输出不失真的合适输入电压, 使输出空载, 把输入、输出

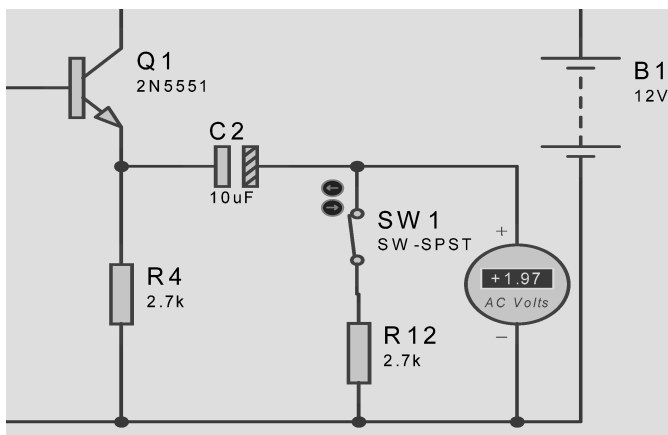


图 7-26 射极跟随器输出电阻的测量

分别接到示波器的 A、B 通道，调节示波器的扫描旋钮和 A、B 通道的垂直位移及增益旋钮，保持两通道的增益一致，垂直位移稍有不同，否则两波形将重叠。观察到的波形如图 7-27 所示。由此可验证射极跟随器名称的由来。

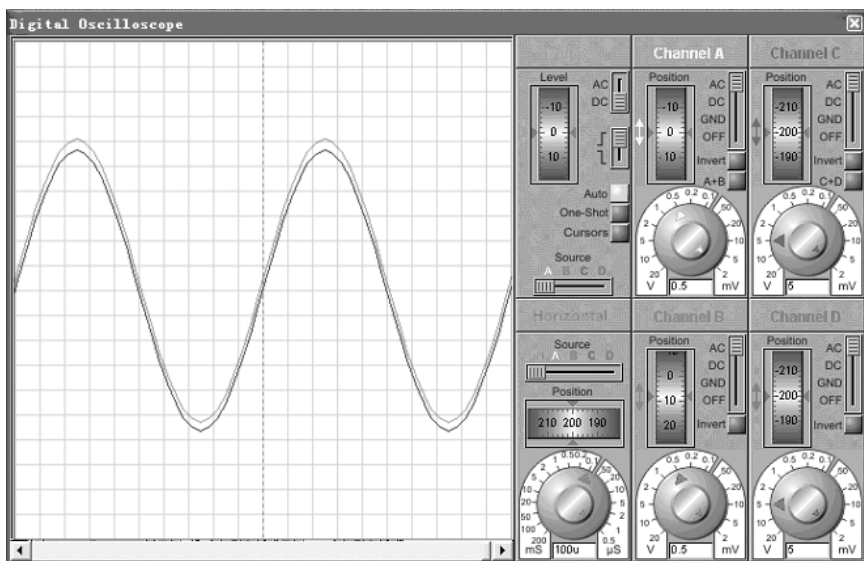


图 7-27 射极跟随器的输入、输出波形



## 7.4 差动放大器

差动放大器用在多级放大电路的第一级，主要目的是减少零漂。与单管共射放大电路相比，差动放大器使用了双倍的元件却得到同样的电压放大倍数，但它具有相当高的共模抑制比，即对共模信号的放大倍数近似为零。差动放大器的实验主要测量电路的静态工作点、单端和双端输出时的差模电压放大倍数  $A_d$ 、共模电压放大倍数  $A_c$  及共模抑制比  $K_{CMRR}$ 。

图 7-28 是差动放大器的实验电路，其中 T3、R1、R2、RE3 构成恒流源，T3 的集电极电流为恒流源的输出。两位开关 K 用来选择差动放大器射极接电阻还是接恒流源，K 拨到左边，

差动放大器接  $10\text{k}\Omega$  的射极电阻  $R_E$ ，拨到右边接恒流源。 $R_W$  是调零电阻，因为在仿真时，我们可以做到差动对管及相应的元件完全对称，而在实际电路中却不能，所以利用调零电阻  $R_W$  来调节两个共射放大电路的对称性。开关  $SW1$  用来在测静态工作点时短接信号源。

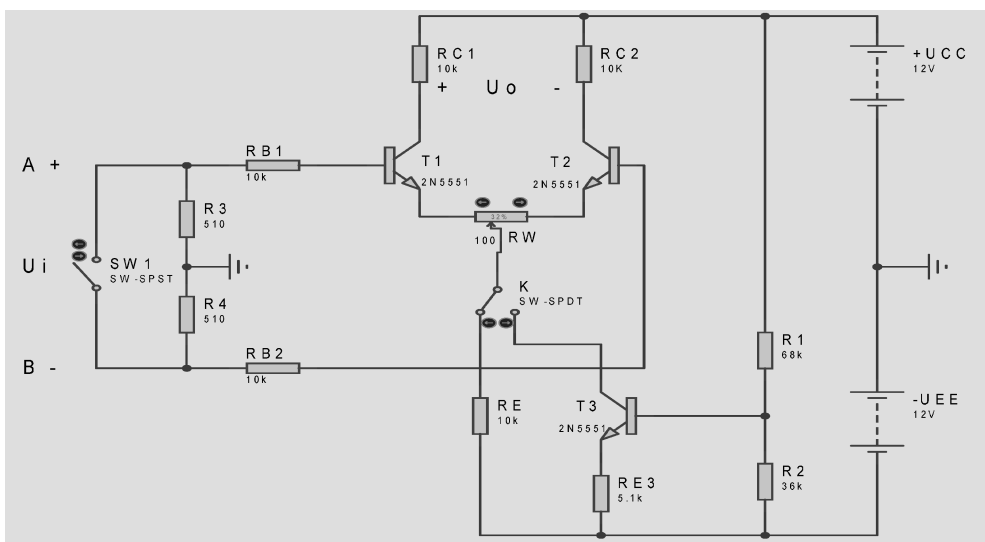


图 7-28 差动放大器实验电路

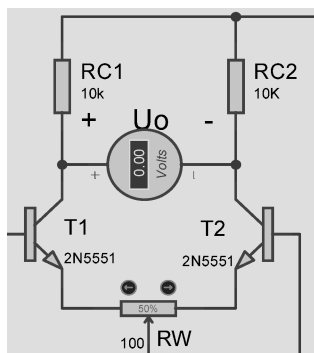


图 7-29 调零电路

### 1. 电路调零

在测各参数之前，先进行电路调零。如图 7-29 所示，在  $T1$ 、 $T2$  管两集电极之间接一直流伏特表，闭合开关  $SW1$ ，把开关  $K$  拨到左边，这时电路中全部为直流量。调节滑动变阻器  $R_W$ ，使电表的读数接近零为止。调零完成，去掉电压表，保持  $R_W$  的触头位置不变。

### 2. 测量静态工作点

在只有直流电源作用的情况下，测得电路中的基极电位、射极电位、集电极电位和集电极电流。

按照图 7-30 连接电路，测得的数据如表 7-3 所示。

表 7-3 差动放大器静态工作点测量值

$V_B/V$	$V_E/V$	$V_C/V$	$I_C/mA$	$V_{BE}/V$	$V_{CE}/V$
-0.02	-0.68	6.37	1.13/2	0.66	7.05

### 3. 单端输出时的放大倍数和共模抑制比

**1) 单端输出差模电压放大倍数** 打开  $SW1$ ，在差模输入端接一信号源，并联交流毫伏表，运行仿真，调节信号源的频率为  $1\text{kHz}$ ，调节信号源的幅值使交流毫伏表的读数约为  $200\text{mV}$ 。在  $T1$  管的集电极接一交流伏特表，如图 7-31 所示。最好在  $T1$  管集电极接示波器，观察输出电压波形不失真为准。

先把开关  $K$  拨到左边，测得  $T1$  管的集电极输出电压为  $7.79\text{V}$ ；再把开关  $K$  拨到右边，

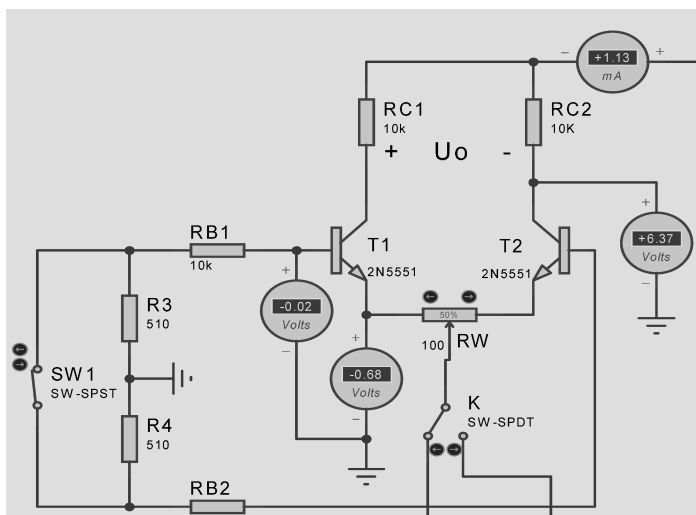


图 7-30 差动放大器的静态工作点测量电路

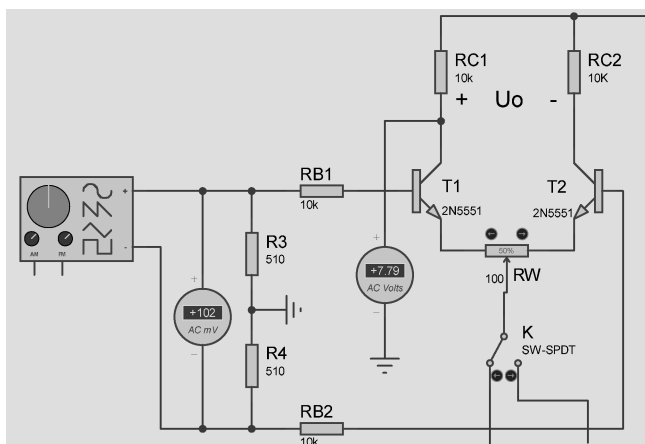


图 7-31 差动放大器的单端输出差模电压放大倍数测量电路

测得 T1 管的集电极输出电压为 6.55V。可计算出射极分别接电阻和恒流源时的单端输出差模电压放大倍数为：

$$A_{d1} = -\frac{U_o}{U_d} \approx -\frac{7.79}{0.1} = -77.9$$

$$A_{d2} = -\frac{U_o}{U_d} \approx -\frac{6.55}{0.1} = -65.5$$

**2) 单端输出共模电压放大倍数** 如图 7-32 所示，把 T1、T2 管的两输入端并联，再接一频率为 1kHz、有效值约为 100mV 的共模输入信号。

先把开关 K 拨到左边，测得 T1 管的集电极输出电压为 6.38V；再把开关 K 拨到右边，测得 T1 管的集电极输出电压为 4.71V。可分别计算出射极接电阻和恒流源时的单端输出共模电压放大倍数为：

$$A_{c1} = -\frac{U_o}{U_c} \approx -\frac{6.38}{0.1} = -63.8$$

$$A_{c2} = -\frac{U_o}{U_c} \approx -\frac{4.71}{0.1} = -47.1$$

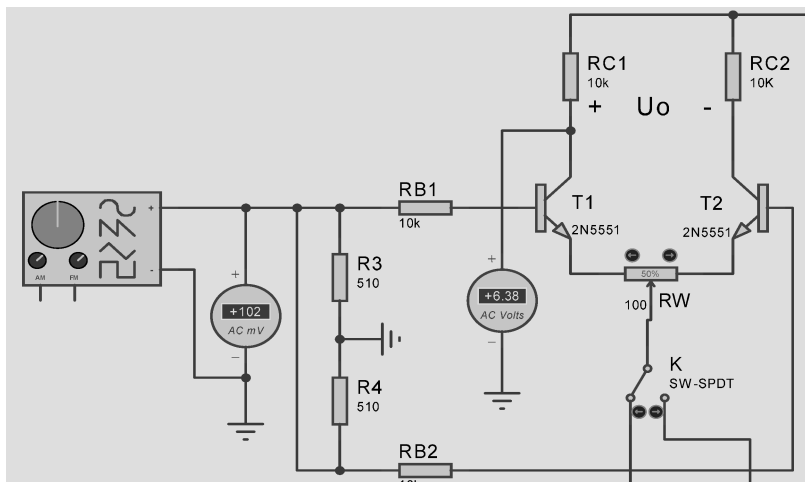


图 7-32 差动放大器的单端输出共模电压放大倍数测量电路

计算结果表明,单端输出时的共模电压放大倍数小于差模电压放大倍数。理想情况下,由于射极电阻较大,共模抑制能力强,共模电压放大倍数应接近零,但这里无论接  $10\text{k}\Omega$  的射极电阻还是接恒流源,共模电压放大倍数都不够理想,即共模电压放大倍数并没有降下来。 $T_1$  和  $T_2$  管接射极电阻和接恒流源两种情况下,由于射极电阻较大,共模抑制能力强,共模放大倍数应接近零,但这里无论接  $10\text{k}\Omega$  的射极电阻还是接恒流源,共模电压放大倍数都不够理想,即共模电压放大倍数并没有降下来。但  $T_1$  和  $T_2$  管接射极电阻和接恒流源两种情况下,第二种接法共模电压放大倍数要小些,效果更好些。这只是测试电路,日常应用中的共模输入电压一般来自温度或其他因素,非故意加之,而是系统输入中所不能剔除的部分,通过差动电路来抑制。

3) 单端输出时的共模抑制比 把单端输出时的差模电压放大倍数比上共模电压放大倍数, 它们的绝对值即共模抑制比, 能反映一个电路对共模信号的抑制能力, 此值越大越好。

可求得本实验电路接射极电阻和接恒流源时的共模抑制比分别为:

$$K_{\text{CMRR1}} = \left| \frac{U_d}{U_c} \right| = \frac{77.9}{63.8} = 1.22$$

$$K_{\text{CMRR2}} = \left| \frac{U_d}{U_c} \right| = \frac{65.5}{47.1} = 1.39$$

#### 4. 双端输出时的放大倍数和共模抑制比

按照前面介绍的差模输入信号和共模输入信号的接法,在输入端分别接 1kHz、有效值约为 100mV 的差模和共模输入信号,在 T1 和 T2 管的集电极之间接一交流电压表,测得双端输出时的差模输出电压和共模输出电压分别为 7.26V 和 0V,如图 7-33 所示。

可算出双端输出时的差模电压放大倍数、共模电压放大倍数及共模抑制比分别为:

$$A_d = -\frac{U_o}{U_d} \approx -\frac{7.26}{0.1} = -72.6$$

$$A_{\tau} = -\frac{U_o}{U_c} \approx -\frac{0}{0.1} = 0$$

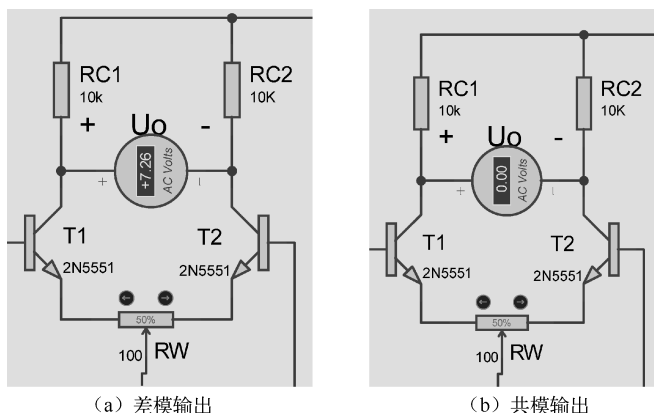


图 7-33 差动放大器的双端输出测量电路

$$K_{\text{CMRR}} = \left| \frac{U_d}{U_c} \right| = \frac{72.6}{0} = \infty$$

可见，双端输出时的共模抑制能力最强。而我们平时所见的电路大部分在差动电路后面还要接单端输入电路，故单端输出应用比较多，这就要求射极电阻足够大，最好接理想恒流源，它的电阻接近 $\infty$ 。

### 5. 输出波形的观察

在差模输入时，如果输入信号的正极性端接 T1 管的基极，由于共射电路的倒相性，单端输出从 T1 管的集电极对地的输出电压是和输入差模信号倒相的；相反，对于同样的输入信号，从 T2 管的集电极输出电压是和输入电压同相的。如图 7-34 所示，分别是单端输出时的两个输出电压及差模输入电压，请大家判断分别是哪个波形。

双端输出时，如果选择 T1 管的集电极为输出电压的正极性端，则输出电压与输入电压同相，否则反相。

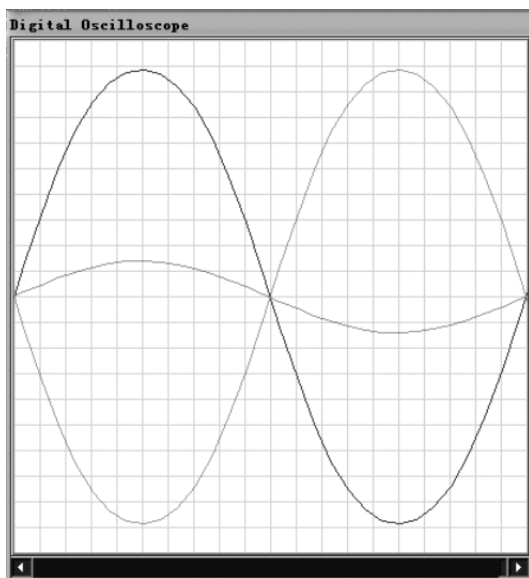


图 7-34 单端输出时的电压波形



## 7.5 思考与练习

1. 请选择合适的运放并设计一个同相放大器，要求其增益为 10 倍，信号源内阻为  $100\text{M}\Omega$ 、信号幅值为  $10\text{mV}$ 。
2. 请选择合适的运放并设计一个反向放大器，要求其增益为 100 倍，由于运放开环增益有限所引起的增益误差小于 1%，信号源内阻为  $100\text{k}\Omega$ 、信号幅值为  $10\text{mV}$ 。
3. 试设计一个带阻滤波器，要求它的 3dB 频率为 300Hz 和 3 000Hz，在 600Hz 和 1 500Hz 之间衰减大于 12dB。
4. 设计一个低通滤波器，要求其通带截止频率  $\omega_p = 1\,000\text{Hz}$ ，通带内增益  $K_p = 1\,000$  且保持平坦，在阻带截止频率  $\omega_r = 1\,500\text{Hz}$  处至少衰减 20dB。
5. 请选择合适的运放并设计一个差动放大器，要求其增益为 30 倍，信号源内阻为  $50\text{k}\Omega$ 、信号幅值为  $20\text{mV}$ 。
6. 请选择合适的运放并设计一个两级放大器，要求前级增益为 5 倍，后级增益为 11 倍，信号源内阻为  $50\text{k}\Omega$ 、信号幅值为  $20\text{mV}$ 。

# 第 8 章 基于 PROTEUS ISIS 的数字电路的仿真与分析

PROTEUS ISIS 的数字电子电路分析支持 JDEC 文件的物理器件仿真，有全系列的 TTL 和 CMOS 数字电子电路仿真模型，可对数字电子电路进行一致性分析。



## 8.1 基于 PROTEUS ISIS 的分立元件门电路的设计与分析

### 1. 与门

#### 1) PROTEUS ISIS 与门电路的设计

(1) 与门。与门（AND Gate）是一种数字电路，只有当所有的输入都是逻辑高电平时，其输出才是逻辑高电平。与门是一种“判决”电路，当所有指定的条件都满足（即为真）时，结果为真，输出为逻辑高电平。

(2) 与门电路图。如图 8-1 所示为与门电路图。

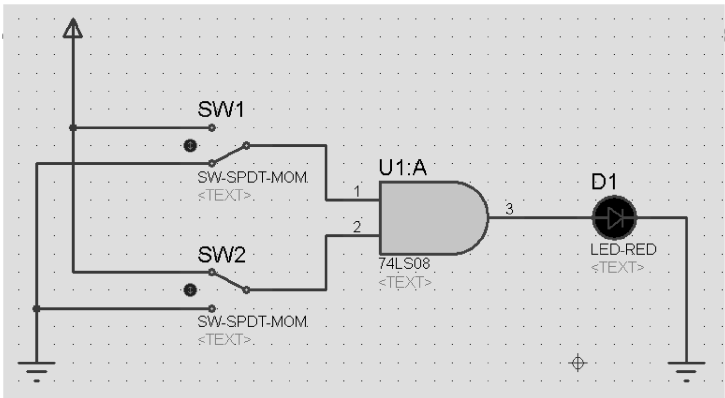


图 8-1 与门电路图

(3) 与门的运算规则。与门的输入个数决定了可能的输入状态组合数，并定义为 2 的幂：  
输入状态组合数 =  $2^n$

其中， $n$  是与门的输入个数。例如，2 输入门有  $2^2 = 4$  种输入状态组合，4 输入门有  $2^4 = 16$  种输入状态组合。

通常使用真值表（Truth Table）来描述与门的逻辑运算。真值表列出所有可能的输入状态组合，并给出相应的输出。真值表用高电平和低电平或 1 和 0 来表示输入和输出状态。

表 8-1 给出了 2 输入与门的真值表，2 输入与门有  $2^2 = 4$  种输入状态组合。其中 1、2 表示门的输入，3 表示门的输出。如表中所示，只有当所有输入都为高电平（1）时，与门输



出才是高电平 (1)；否则输出为低电平 (0)。

表 8-1 2 输入与门的真值表

输 入		输 出
1	2	3
低 (0)	低 (0)	低 (0)
低 (0)	高 (1)	低 (0)
高 (1)	低 (0)	低 (0)
高 (1)	高 (1)	高 (1)

## 2) PROTEUS ISIS 与门电路的调试与仿真

(1) 元器件的选择及放置。點選元件模式 (Component Mode) 图标, 单击 “P” 按钮, 从弹出的选取元件对话框中选择与门电路的仿真元件, 仿真元件信息如表 8-2 所示。

表 8-2 仿真元件信息表

元 件 名 称	所 属 类	所 属 子 类
SW-SPDT-MOM (单刀双掷开关)	Active	Switch
74LS08 (与门)	74LS	74LS
LED-RED (发光二极管)	Active	LED

将仿真元件添加到对象选择器后关闭元件选取对话框。

放置仿真元件, 结果如图 8-2 所示。

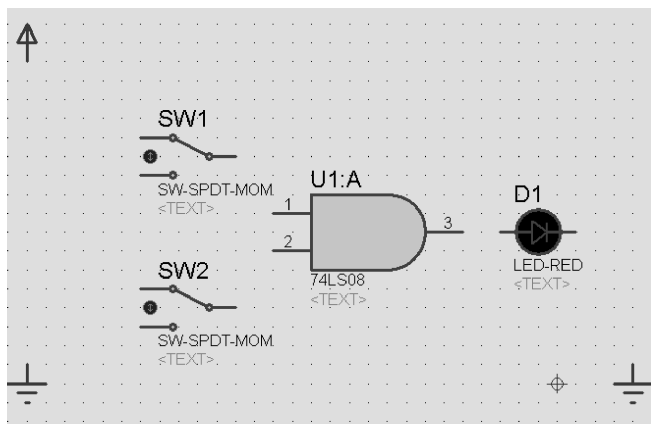


图 8-2 与门仿真电路图

根据与门电路图连线。

(2) PROTEUS ISIS 的仿真与分析。单击仿真按钮, 仿真结果如图 8-3 所示。

## 2. 或门

### 1) PROTEUS ISIS 或门电路的设计

(1) 或门。或门 (OR Gate) 是一种数字电路, 只要有一个或多个输入是逻辑高电平, 其输出就是逻辑高电平。或门也是一种 “判决” 电路, 只有一个或多个输入条件都满足 (即为真) 时, 结果就为真, 输出逻辑高电平。

(2) 或门电路图。如图 8-4 所示为或门电路图。

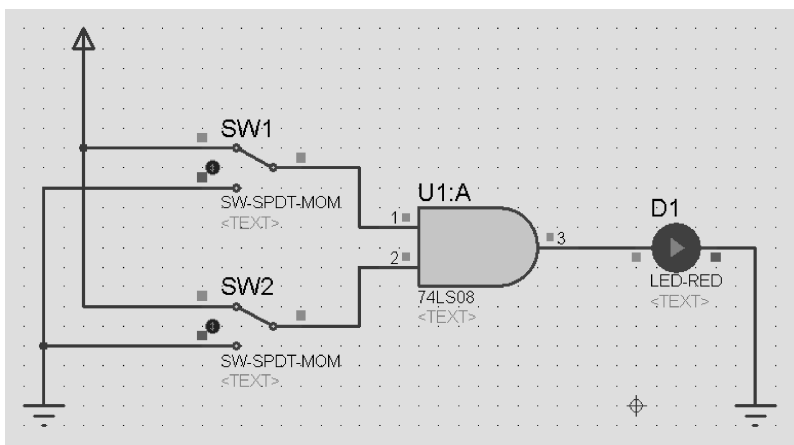


图 8-3 与门电路仿真结果

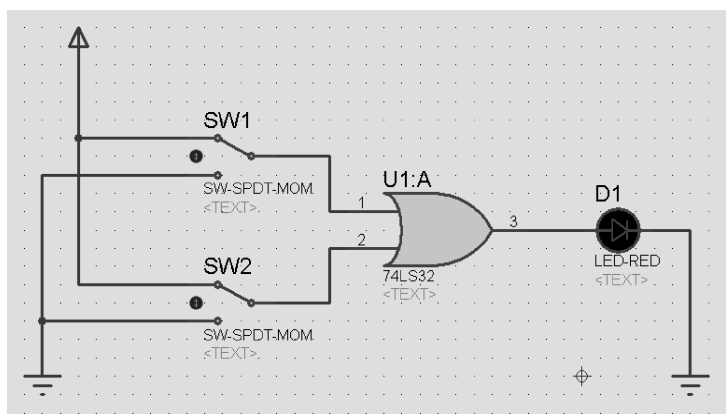


图 8-4 或门电路图

(3) 或门的运算规则。类似于与门，或门的输入个数决定了可能的输入状态组合数，并定义为 2 的幂：

$$\text{输入状态组合数} = 2^n$$

其中， $n$  是与门的输入个数。

可以使用真值表来描述或门的逻辑运算。真值表列出所有可能的输入状态组合，并给出相应的输出。真值表用高电平和低电平或 1 和 0 来表示输入和输出状态。

表 8-3 给出了 2 输入或门的真值表，2 输入或门有  $2^2 = 4$  种输入状态组合。其中 1、2 表示门的输入，3 表示门的输出。如表中所示，只要有一个或多个输入为高电平（1），或门的输出就是高电平（1）。

表 8-3 2 输入或门的真值表

输 入		输 出
1	2	3
低 (0)	低 (0)	低 (0)
低 (0)	高 (1)	高 (1)
高 (1)	低 (0)	高 (1)
高 (1)	高 (1)	高 (1)

## 2) PROTEUS ISIS 或门电路的调试与仿真

(1) 元器件的选择及放置。点选元件模式 (Component Mode) 图标, 单击 “P” 按钮, 从弹出的选取元件对话框中选择与门电路的仿真元件, 仿真元件信息如表 8-4 所示。

表 8-4 仿真元件信息表

元件名称	所属类	所属子类
SW-SPDT-MOM (单刀双掷开关)	Active	Switch
74LS32 (或门)	74LS	74LS
LED-RED (发光二极管)	Active	LED

将仿真元件添加到对象选择器后关闭元件选取对话框。

放置仿真元件, 结果如图 8-5 所示。

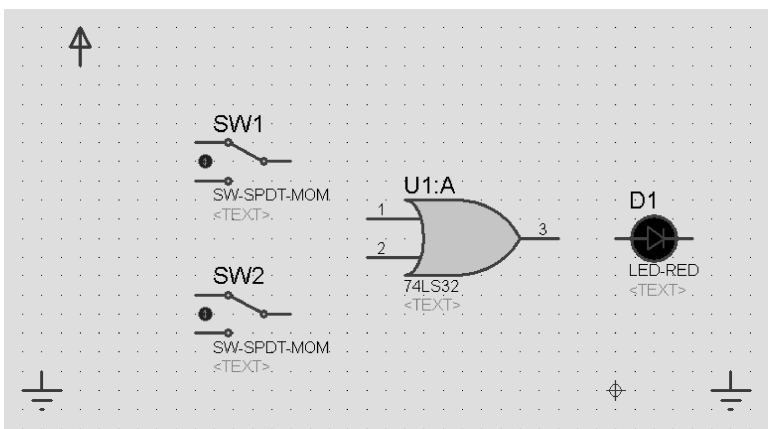


图 8-5 或门仿真电路图

根据或门电路图连线。

(2) PROTEUS ISIS 的仿真与分析。或门电路仿真结果如图 8-6 所示。

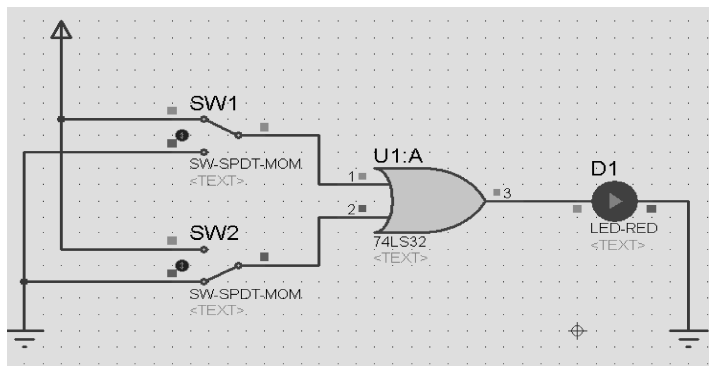


图 8-6 或门电路仿真结果

## 3. 非门

### 1) PROTEUS ISIS 非门电路的设计

(1) 非门。非门输出的逻辑电平与输入的逻辑电平正好相反, 非门也称为非电路。

反相 (Inversion) 是一种对逻辑电平取反的运算。非门 (Inverter, 也称为反向器) 是一种逻辑电路, 只有一个输入和一个输出。非门将低电平转变为高电平, 将高电平转变为低电平。对于二进制数而言, 非门将 0 转变为 1, 将 1 转变为 0。非门也称为非 (NOT) 电路, 因为其输出状态与输入不同。

(2) 非门电路图。如图 8-7 所示为非门电路图。

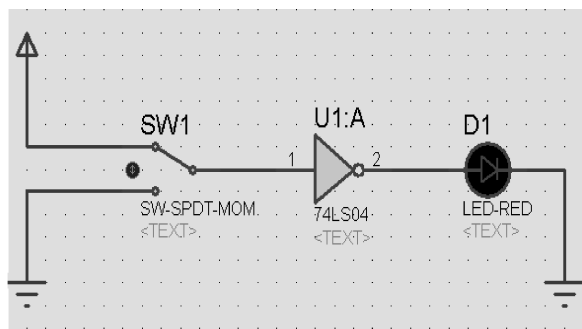


图 8-7 非门电路图

(3) 非门运算规则。断言状态 (Asserted State) 也称为有效状态 (Active State), 断言状态是指设计者认为逻辑门或非门应该处于的逻辑状态或电平 (高或低)。例如, 假设使用非门的输出来驱动 LED (发光二极管), 需要输出低电平。因此, 低电平就是非门输出的断言状态。为了得到低电平输出, 非门的输入必须是高电平, 因此高电平是非门输入状态的断言状态。既然这样, 小圆圈 (即取反指示符) 应该位于非门的输出端。当然, 当输入为低电平时, 输出为高电平, 此时为非断言 (Nonasserted) 状态, LED 熄灭。

非门的逻辑运算可以用简单的真值表来描述, 见表 8-5。其中, 1 表示非门的输入, 2 表示非门的输出。

表 8-5 非门的真值表

输 入	输 出
1	2
低 (0)	高 (1)
高 (1)	低 (0)



## 8.2 基于 PROTEUS ISIS 的组合逻辑电路的设计与仿真

### 1. 与非门

#### 1) PROTEUS ISIS 与非门电路设计

(1) 与非门电路。在基本逻辑组合门电路中与非门电路最为常用, 其逻辑功能为: 当输入变量全为 1 时, 输出为 0, 当输入变量有一个或几个为 0 时, 输出为 1。简言之, 即全为 1 出 0, 有 0 出 1。

(2) 与非门电路图。如图 8-8 所示为与非门电路图。

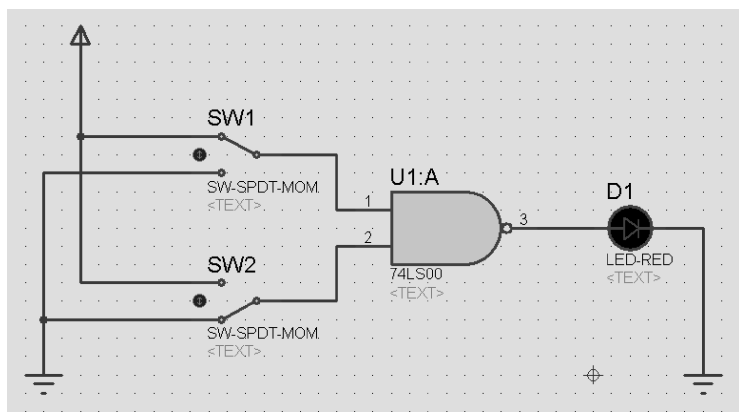


图 8-8 与非门电路图

(3) 与非门的运算规则。本次仿真是用与非门电路驱动发光二极管，当与非门元器件两端输出为 1（高电平）时发光二极管亮，反之二极管灭。与非门的运算规则可以用真值表来表示，见表 8-6。

表 8-6 与非门真值表

SW1	SW2	输 出
0	0	1
0	1	1
1	0	1
1	1	0

## 2) PROTEUS ISIS 调试与仿真

(1) 元器件的选择与放置。點選元件模式（Component Mode）图标，单击“P”按钮，从弹出的选取元件对话框中选择与非门电路的元器件，具体如表 8-7 所示。

表 8-7 与非门仿真电路所用元器件

元 件 名 称	所 属 类	所 属 子 类
SW-SPDT-MOM	Active	Switch
74LS00（与非门）	74LS	74LS
LED-RED	Active	LED

按图 8-9 所示放置元器件，并接入电源和地。

根据与非门电路图连线，并进行电气检查。

(2) PROTEUS ISIS 的仿真与分析。根据真值表依次执行，观察灯的亮灭情况，以验证真值表的正确性。电路仿真结果如图 8-10 所示。

## 2. 或非门

### 1) 或非门电路的电路设计

(1) 或非门。或非门的逻辑功能为：输入变量全为 0 时，输出为 1，当输入变量有一个或一个以上为 1 时输出为 0，即全 0 为 1，有 1 为 0。

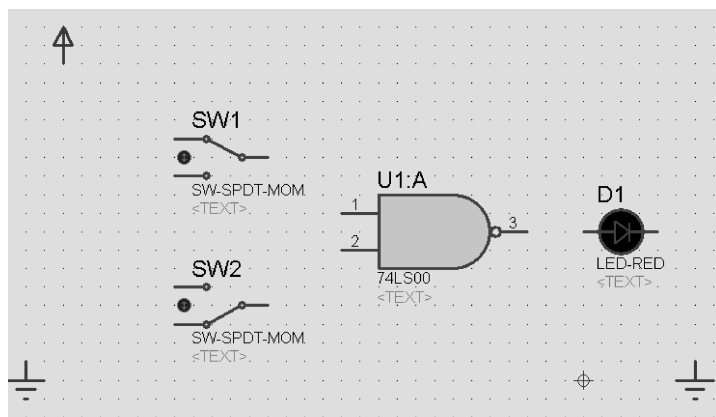


图 8-9 与非门仿真电路图

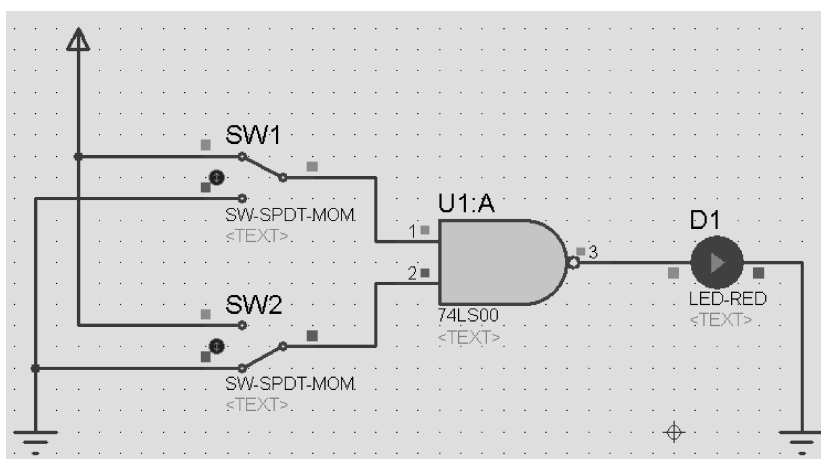


图 8-10 与非门电路仿真结果

(2) 或非门的电路图。如图 8-11 所示为或非门电路图。

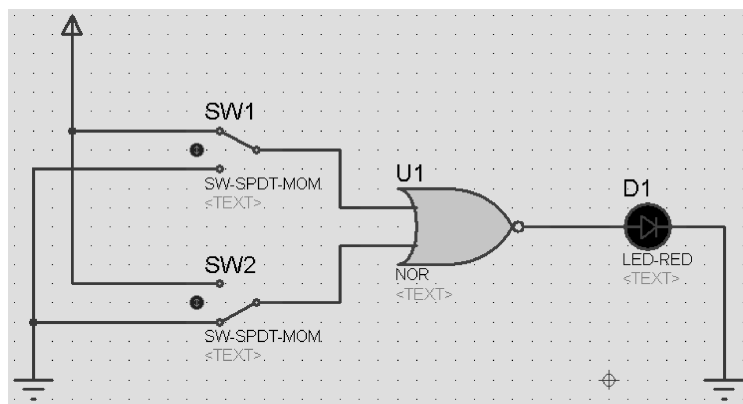


图 8-11 或非门电路图

(3) 或非门的运算规则。本次仿真用或非门电路驱动发光二极管，当或非门元器件两端输出为 0（低电平）时发光二极管亮，反之二极管灭。或非门的运算规则可以用真值表来表示，见表 8-8。

表 8-8 或非门真值表

SW1	SW2	输 出
0	0	1
0	1	0
1	0	0
1	1	0

## 2) PROTEUS ISIS 调试与仿真

(1) 元器件的选择与放置。点选元件模式 (Component Mode) 图标, 单击 “P” 按钮, 从弹出的选取元件对话框中选择或非门电路的元器件, 具体如表 8-9 所示。

表 8-9 或非门仿真电路所用元器件

元 件 名 称	所 属 类	所 属 子 类
SW-SPDT-MOM	Active	Switch
NOR (或非门)	Active	Simple
LED-RED	Active	LED

按图 8-12 所示放置元器件, 并接入电源和地。

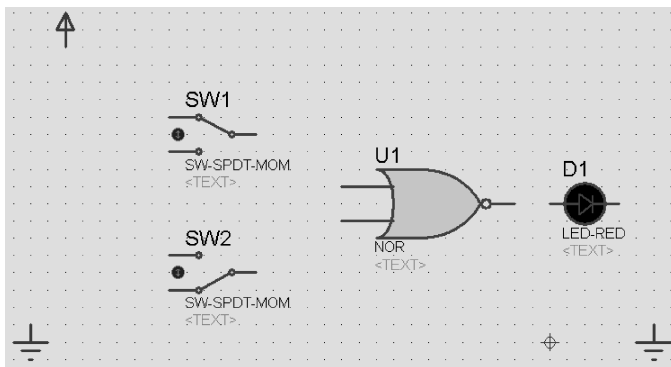


图 8-12 或非门仿真电路图

根据或非门电路图连线, 并进行电气检查。

(2) PROTEUS ISIS 的仿真与分析。根据真值表依次执行, 观察灯的亮灭情况, 以验证真值表的正确性。电路仿真结果如图 8-13 所示。

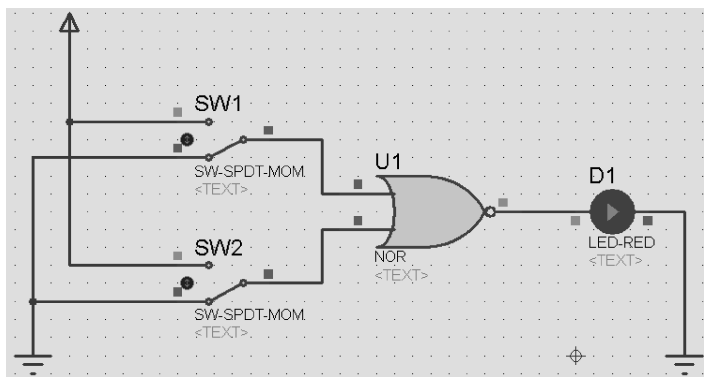


图 8-13 或非门电路仿真结果



## 8.3 基于 PROTEUS ISIS 的编码器电路的设计与仿真

十进制数码或某种文字和符号的编码难以用电路实现。在数字电路中,一般用的是二进制编码。二进制只有 0 和 1 两个数码,可以把若干个 0 和 1 按一定规律排起来组成不同的代码(二进制数),来表示某一对象或信号。1 位二进制代码有 0 和 1 两种,可以表示两个信号;2 位二进制代码有 00、01、10、11 四种,可以表示四个信号。 $n$  位二进制代码有  $2^n$  种,可以表示  $2^n$  个信号。这种二进制编码在电路上容易实现。下面讨论优先制编码器。

优先编码器的功能是允许同时在几个输入端有输入信号,编码器按输入信号预先排定的优先顺序,只对同时输入的几个信号中优先权最高的一个信号编码。信号优先权级别的高低是根据各信号的轻重缓急人为设定的。例如,计算机有许多输入设备,可能多台设备同时向主机发出中断请求,希望输入数据,这就要求主机能自动识别这些请求信号的优先级别,按次序进行编码。这里就需要优先编码器。74LS147 型 10-4 线优先编码器是常用的优先编码器,表 8-10 是其真值表。由表可见,有 9 个输入变量  $\bar{I}_1 \sim \bar{I}_9$ , 4 个输出变量  $\bar{Y}_3 \sim \bar{Y}_0$ , 它们都是反变量,表明它们均为低电平有效,即有信号时,输入为 0。输出的反变量组成 8421 BCD 的反码,对应于 0~9 十个十进制数码。输入信号的优先次序为  $\bar{I}_9 \sim \bar{I}_1$ 。例如,当  $\bar{I}_9=0$  时,无论其他输入端是 0 还是 1,输出端都只能对  $\bar{I}_9$  编码,输入为 0110 (原码为 1001)。

表 8-10 编码器真值表

输 入									输 出			
$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{I}_8$	$\bar{I}_9$	$\bar{Y}_3$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$
H	H	H	H	H	H	H	H	H	H	H	H	H
x	x	x	x	x	x	x	x	L	L	H	H	L
x	x	x	x	x	x	x	L	H	L	H	H	H
x	x	x	x	x	x	L	H	H	H	L	L	L
x	x	x	x	x	L	H	H	H	H	L	L	H
x	x	x	x	L	H	H	H	H	H	L	H	L
x	x	x	L	H	H	H	H	H	H	L	H	H
x	x	L	H	H	H	H	H	H	H	H	L	L
x	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

说明: H—高电平; L—低电平; x—任意

### 1. PROTEUS ISIS 电路设计

这次的编码器采用 10-4 线优先编码器 (BCD 输出) 编码输出, 经过非门反向后用 4 位数码管显示, 电路图如图 8-14 所示。

按照电路图在 PROTEUS 中选择元器件, 元器件清单如表 8-11 所示。

根据电路图放置元器件, 电阻阻值为  $10\text{k}\Omega$ , 连接电路图, 并进行电气检测。



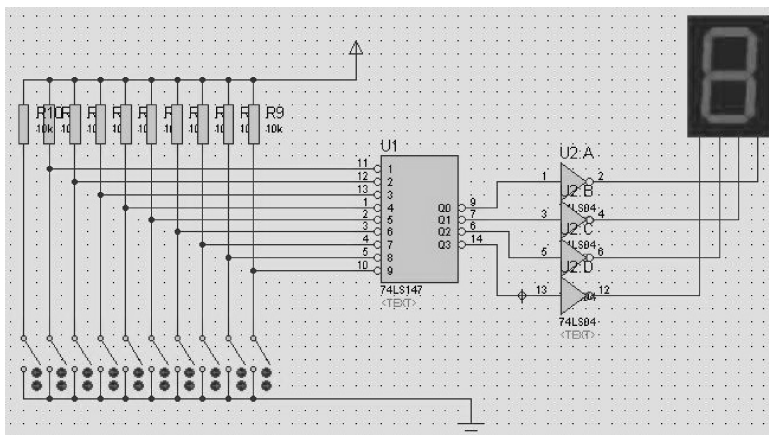


图 8-14 编码器电路图

表 8-11 编码器仿真电路所用元器件

名 称	所 属 类	所 属 子 类
74LS147	74LS	74LS
RES	Device	Generic
SW-SPDT-MOM	Active	Switch
74LS04	74LS	Hex inverters
74EG-BCD	Display	Display

## 2. PROTEUS ISIS 调试与仿真

单击 PROTEUS ISIS 界面左下角的仿真按钮，观察数码管的显示及 74LS147 各引脚的电平。其实此电路就是十键 8421 码编码器的逻辑图，按下某个按键，输入相应的一个十进制数码。例如，按下  $S_5$  键（电路图中按键顺序自左向右为  $S_0 \sim S_9$ ），输入为 5，即  $\bar{I}_5 = 0$ ，输出为 0101；按下  $S_0$  键，则输出为 0000。依次按下每个按键，则顺序显示 0、1、2、3、4、5、6、7、8、9。图 8-15 即为仿真效果图（此时按下的是  $S_8$  键）。

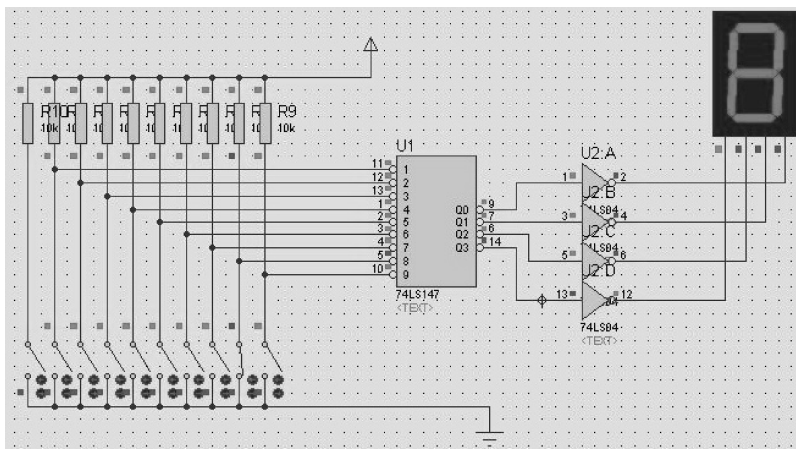


图 8-15 编码器电路仿真结果



## 8.4 基于 PROTEUS ISIS 的译码器电路的设计与仿真

译码与编码的过程相反。编码是将某种信号或十进制的 10 个数码（输入）编成二进制代码（输出）；译码是将二进制代码（输入）按其编码时的原意译成对应的信号或十进制数码（输出）。

在 8.3 节用到的数码管就是一种显示译码器，将二进制数译为十进制数并显示，本节将介绍二 - 十进制显示译码器的 7 段显示译码器。

7 段显示译码器的功能是把“8421”二 - 十进制代码译成对应于数码管的 7 个字段的信号，驱动数码管，显示相应的十进制数码。如果是 74LS247 型译码器，输出低电平有效，应采用共阳极数码管，则 7 段显示译码器的功能表如表 8-12 所示；如果是 74LS248 型译码器，输出高电平有效，应采用共阴极数码管，则输出状态表应与表 8-12 相反，即 1 和 0 对换。

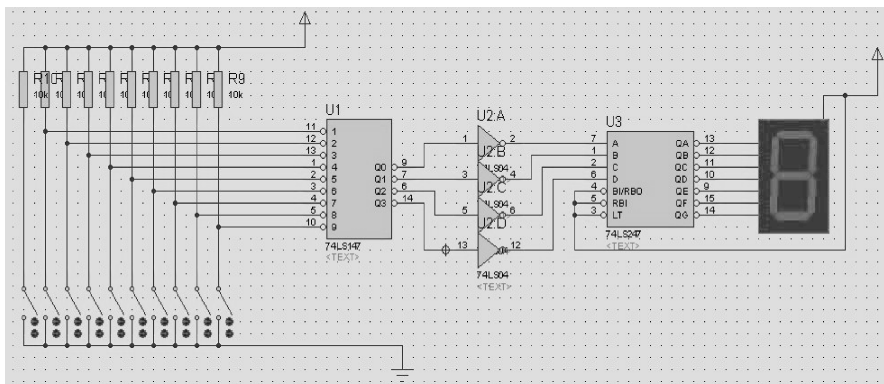
表 8-12 译码器功能表

十进制	输 入						BI	输 出							说明
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	×	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	×	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	×	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	×	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	×	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	×	L	H	H	L	H	ON	OFF	ON	ON	ON	ON	ON	
7	H	×	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	×	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	×	H	L	L	H	H	ON	ON	ON	ON	OFF	ON	ON	
10	H	×	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	×	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	×	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	×	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	×	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	×	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	×	×	×	×	×	×	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	×	×	×	×	×	H	ON	ON	ON	ON	ON	ON	ON	4

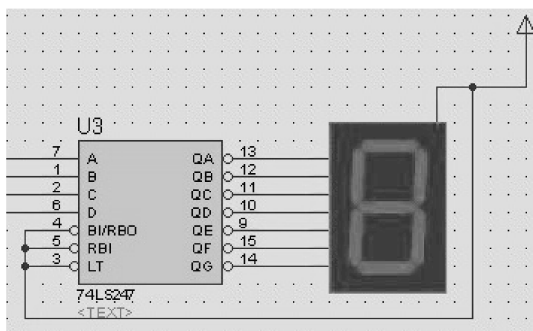
### 1. PROTEUS ISIS 电路设计

译码器电路设计是将 8.3 节中设计的编码器输出电路与 74LS247、共阳极半导体数码管组成的电路，电路图如图 8-16 所示。

按照电路图在 PROTEUS 中选择元器件，元器件清单如表 8-13 所示。



(a) 译码器电路总图



(b) 译码器电路图

图 8-16 译码器电路设计

表 8-13 译码器仿真电路所用元器件

名 称	所 属 类	所 属 子 类
74LS147	74LS	74LS
RES	Device	Generic
SW-SPDT-MOM	Active	Switch
74LS04	74LS	Hex Inverters
74LS247	74LS	74LS
7SEG-COM-ANODE	Display	Display

根据电路图放置元器件，电阻阻值为  $10k\Omega$ ，连接电路图，并进行电气检测。

## 2. PROTEUS ISIS 调试与仿真

单击仿真按钮进行仿真，数码管显示与 8.3 节中的显示是完全一致的。观察仿真电路中数码管及 74LS247 各引脚电平的变化，并做记录。



## 8.5 基于 PROTEUS ISIS 的 D 触发器电路的设计与仿真

D 触发器是一种常用的边沿触发器，它的逻辑状态只取决于 CP 边沿（上升沿或下降沿）到达时刻输入信号的状态，而与此边沿时刻以前或以后的输入状态无关，因而可以提

高它的可靠性和抗干扰能力。

本节介绍的是一种目前用得较多的维持阻塞型 D 触发器，由 6 个与非门组成，其中 G1、G2 组成基本触发器，G3、G4 组成时钟控制电路，G5、G6 组成数据输入电路。其逻辑图如图 8-17 所示。

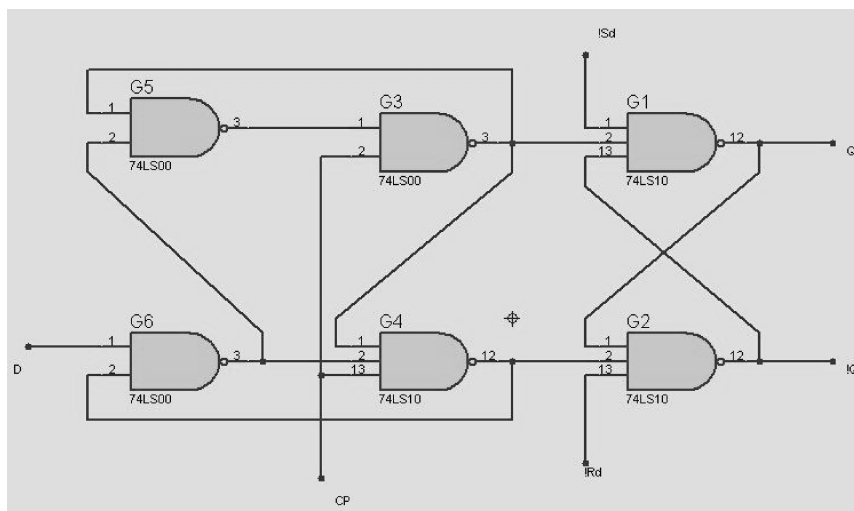


图 8-17 D 触发器逻辑图

## 1. PROTEUS ISIS 电路设计

**1) D 触发器的逻辑功能** 维持阻塞型 D 触发器具有在时钟脉冲上升沿触发的特点，其逻辑功能为：输出端 Q 的状态随着输入端 D 的状态而变化，但总比输入端状态的变化晚一步，即某个时刻的脉冲到来之后 Q 的状态和该脉冲到来之前 D 的状态一样。

所以此次仿真采用 AT89C51 单片机的 P0.0 口作为脉冲输出端，连接 74LS74D 触发器，然后由示波器显示输出波形。图 8-18 是仿真电路图。

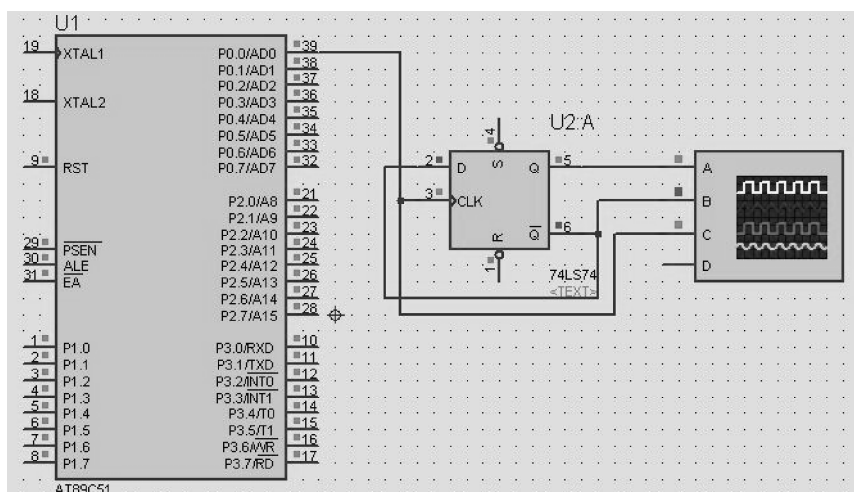


图 8-18 D 触发器仿真电路图

**2) 元器件的选择** 根据电路图选择元器件，如表 8-14 所示。

表 8-14 D 触发器仿真电路所用元器件

名 称	所 属 类	所 属 子 类
74LS74	74LS	D-FLIP-FLOP
AT89C51	MCS8051	8051 MCU

## 2. PROTEUS ISIS 调试与仿真

根据图 8-18 所示电路图绘制电路, 检查无误后, 单击仿真按钮进行仿真。仿真结果可以通过示波器显示, 如图 8-19 所示。

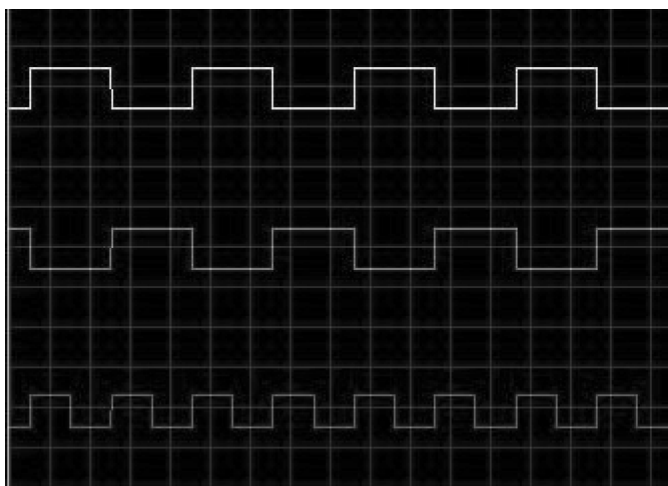


图 8-19 D 触发器仿真结果

图 8-19 中 3 条波形自上向下依次为  $Q$ 、 $D(\bar{Q})$ 、 $CP$ 。由仿真结果可以看出, 只有在脉冲上升沿  $Q$  的输出才改变, 且为  $D(\bar{Q})$  上升沿之前的状态。



## 8.6 思考与练习

1. 设计交通信号灯故障检测电路。
2. 设计两地控制一灯的电路。
3. 某车间有 A、B、C、D 四台电动机, 要求: ①A 机必须开机; ②其他三台电动机至少有两台开机。若不满足上述要求, 则指示灯熄灭。设指示灯亮为“1”, 灭为“0”。电动机的开机信号通过某种装置送到各自的输入端, 使该输入端为“1”, 否则为“0”。试用与非门设计该系统。
4. 试用 74LS175 设计 4 人抢答电路。
5. 试用两片 74LS293 型计数器接成二十四进制计数器。

# 第9章 基于 PROTEUS ISIS 的单片机仿真

在基于微处理器系统设计中，即使没有物理原型，PROTEUS VSM 也能进行软件开发。模型库中包含 LCD 显示、键盘、按钮、开关等通用外设，同时，提供的 CPU 模型有 8051/8052、PIC、AVR、ARM7 等。

单片机系统的仿真是 PROTEUS VSM 的一大特色。而且，该仿真系统将源代码的编辑和编译整合到同一个设计环境中，这样使得用户可以在设计中直接编辑代码，并且很容易地查看到用户对源程序修改后对仿真结果的影响。



## 9.1 单片机程序的建立与编译

PROTEUS VSM 提供了简单的文本编辑器，用它作为源程序的编辑环境。对于不同系列的单片机，VSM 均提供了相应的编译器，使用时可根据单片机的型号和语言进行选择。

编译器有以下几种：

- ASEM51 (51 系列单片机编译器)；
- ASM11 (Motorola 单片机编译器)；
- AVRASM2 (Atmel AVR 系列单片机编译器)；
- MPASM (PIC 单片机编译器)；
- MPASMS (WIN PIC 单片机编译器)。

### 1. 源代码文件的创建

在 PROTEUS ISIS 界面中单击菜单【Source】（源程序），在弹出的下拉菜单中单击【Add/Remove Source Code Files...】（添加/移除源程序）选项，弹出如图 9-1 所示对话框，

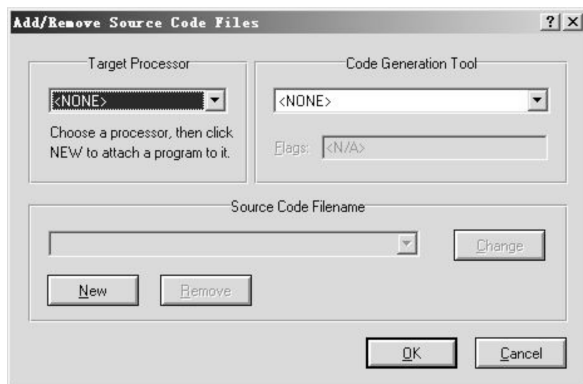



图 9-1 “Add/Remove Source Code Files”对话框

单击“Code Generation Tool”（目标代码生成工具）下方框中的按钮, 弹出下拉菜单, 根据需求选择相应的编译器, 如“ASEM51”（51 系列单片机编译器）。

在图 9-1 所示对话框中单击“New”按钮, 弹出如图 9-2 所示对话框, 在文件名栏中输入新建源程序文件名“test”, 单击“打开”按钮, 弹出如图 9-3 所示对话框, 选择“是”按钮, 新建的源程序文件就添加到图 9-1 的“Source Code Filename”方框中, 如图 9-4 所示。同时在 ISIS 界面的“Source”菜单中也加入了源程序文件名“test. ASM”, 如图 9-5 所示。

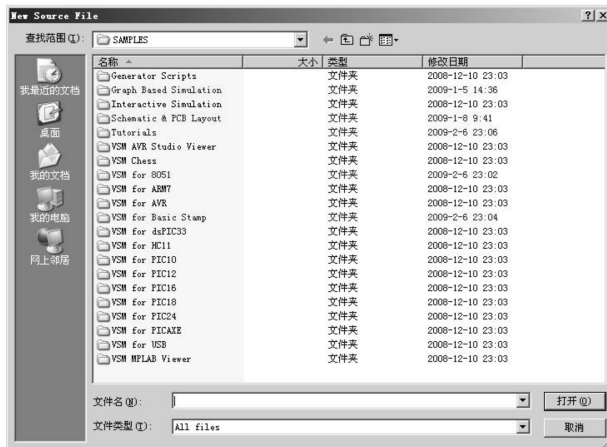


图 9-2 新建源程序文件对话框

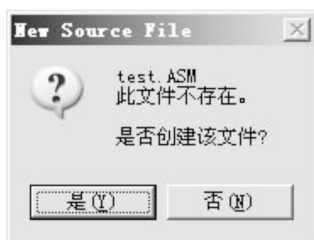


图 9-3 新建文件确定

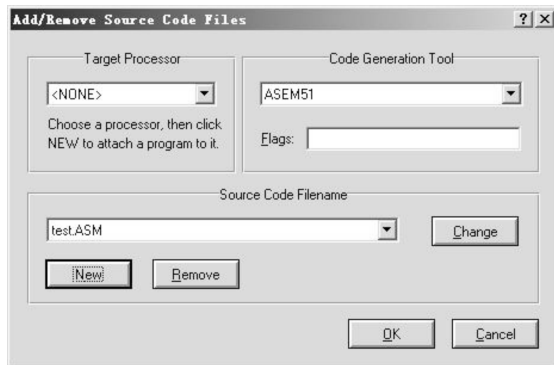


图 9-4 源程序添加结果

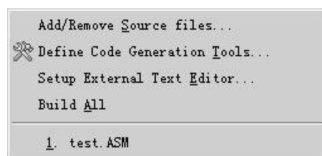


图 9-5 菜单“Source”中加载的源程序文件

## 2. 源代码编辑

单击菜单【Source】→【test. ASM】, 出现如图 9-6 所示的源程序编辑窗口, 编写源程序后保存并退出。

## 3. 编译源代码生成目标代码

1) 编译器设置 第一次使用编译器时需要进行相关的设置, 单击菜单【Source】→【Define Code Generation Tools】, 出现如图 9-7 所示界面, 本例中实际设置结果如图 9-7 所示。

2) 编译源程序, 生成目标代码文件 单击菜单【Source】→【Build All】, 编译结果出现在弹出的编译日志对话框中。如果出现错误, 按照错误提示进行修改; 如果没有错误, 便

成功生成目标代码“.HEX”文件。

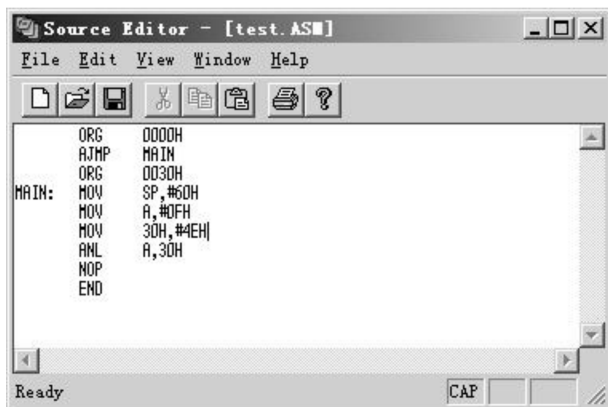


图 9-6 源程序编辑窗口

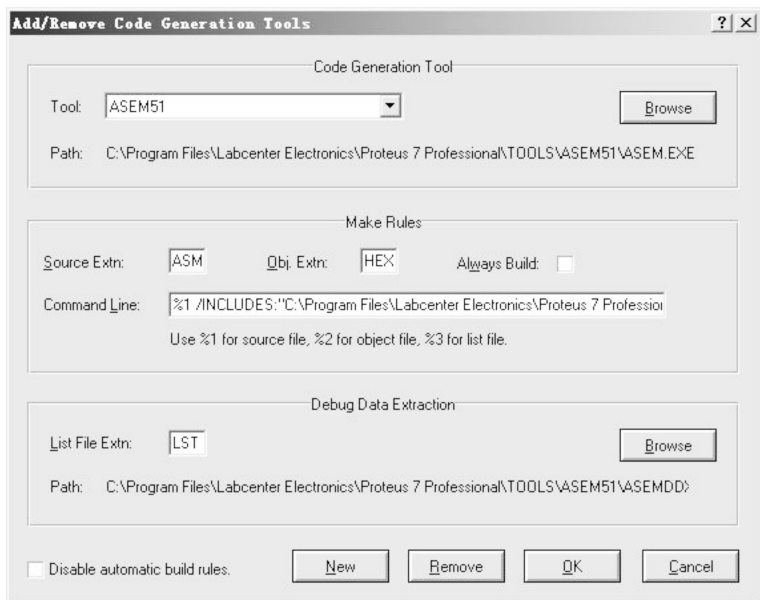


图 9-7 编译器设置界面



## 9.2 单片机电路的仿真与调试

PROTEUS VSM 支持源代码调试，系统的 Debug Loaders 包含在系统文件 LOADERS.DLL 中。目前，系统可支持的工具有数量正在迅速增加。

对于系统支持的汇编程序或编译器，PROTEUS VSM 会为设计项目中的每一个源代码文件创建一个源代码窗口，并且这些代码会在 Debug 菜单中显示。

在进行代码调试时，必须先在微处理器属性编辑的 Program File 项中配置目标代码文件名 [通常为 HEX、S19 或符号调试数据文件 (Symbolic Debug Data File)]。ISIS 不能自动获取目标代码，因为在设计中可能有多个处理器。



## 1. 添加目标代码到电路

在 PROTEUS ISIS 编辑环境中，双击 AT89C51 单片机，弹出如图 9-8 所示的 AT89C51 元件属性编辑对话框。

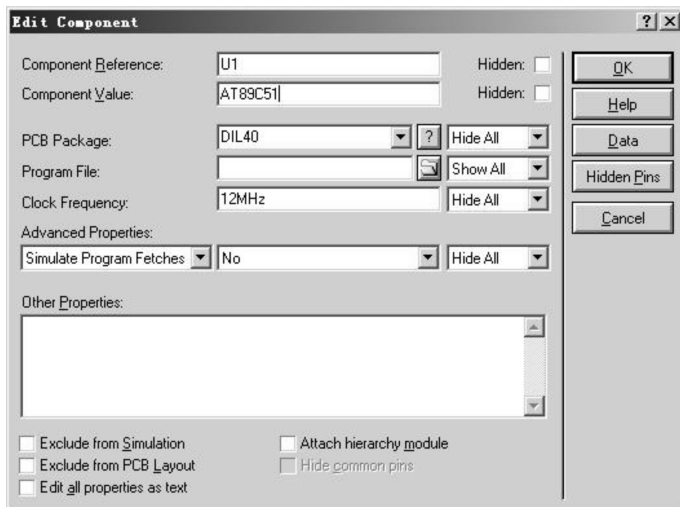


图 9-8 AT89C51 元件属性编辑对话框

单击“Program File”文本框右侧的打开按钮，如图 9-9 所示，弹出如图 9-10 所示的文件浏览窗口。选择需要添加的目标代码文件后，单击“打开”按钮，此时就将目标代码添加到了电路中，如图 9-11 所示。单击“OK”按钮完成编辑。



图 9-9 单击“Program File”文本框右侧的打开按钮

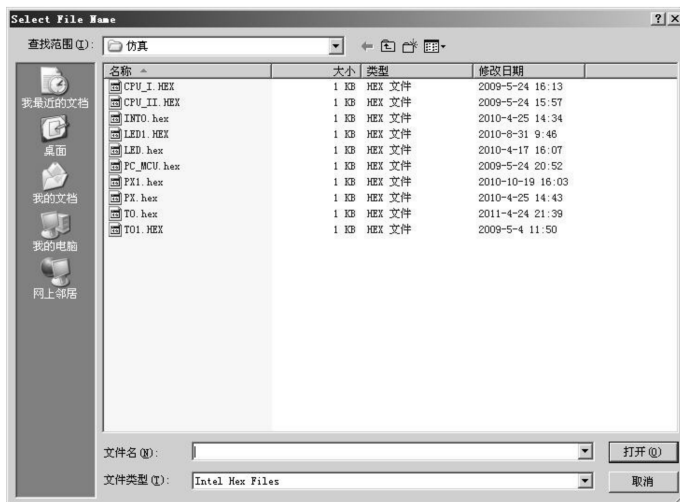


图 9-10 文件浏览窗口

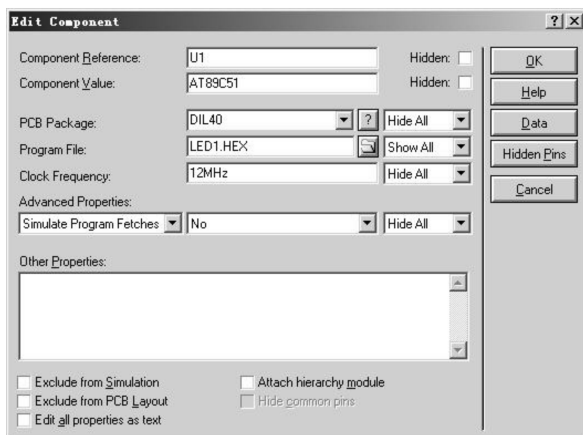




图 9-11 添加目标代码到电路中

## 2. 电路调试

在 PROTEUS 仿真界面中单击按钮 , 全速启动仿真, 此时电路中的 LED 按照不同的花式点亮。单击停止按钮 , 可停止仿真。仿真实例片段如图 9-12 所示。

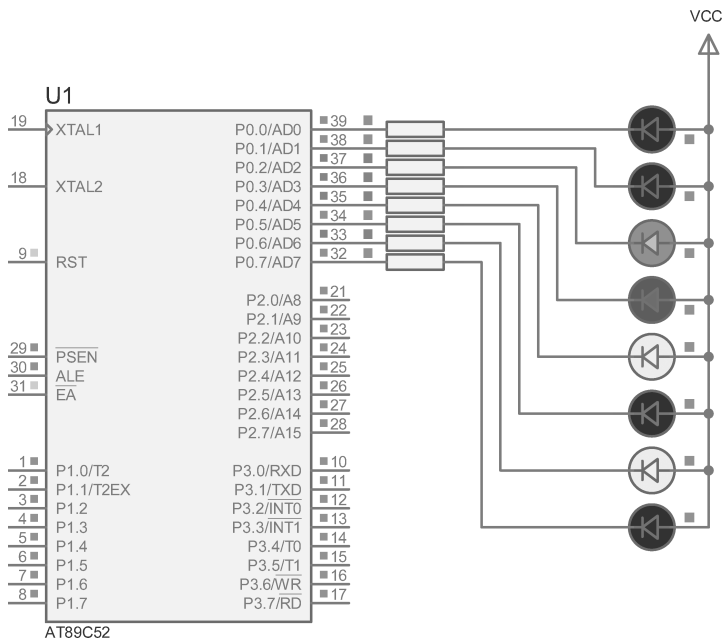


图 9-12 仿真实例片段



## 9.3 PROTEUS ISIS 与 Keil C 的联合调试

Keil 是德国开发的一个微处理器开发软件平台, 可以进行纯粹的软件仿真 (仿真软件程序, 不接硬件电路), 也可以利用硬件仿真器, 搭接上单片机硬件系统, 在仿真器中载入项目程序后进行实时仿真; 还可以使用其内嵌的 Keil Monitor - 51, 在不需要额外的硬件仿真器的条件下, 搭接单片机硬件系统对项目程序进行实时仿真。这样, Keil 就可以很好地与 PROTEUS 结合, 对单片机程序进行实时仿真调试。

## 1. PROTEUS ISIS 与 Keil C 的接口

在 Keil 中调用 PROTEUS 进行单片机外围元器件的仿真，步骤如下。

- (1) 安装 Keil 与 PROTEUS。
- (2) 把 PROTEUS\MODELS 目录中的 VDM51.DLL 复制到 Keil 安装目录的\C51\BIN 目录中。
- (3) 修改 Keil 安装目录下的 TOOLS.INI 文件，在 C51 字段加入 TDRV8 = BIN\VDM51.DLL ( “Proteus VSM Monitor - 51 Driver” ) 并保存，如图 9-13 所示。



图 9-13 修改 Keil 安装目录下的 TOOLS.INI 文件

- (4) 打开 PROTEUS，画出相应电路，在 PROTEUS 的【Debug】菜单中选择【Use Remote Debug Monitor】选项，如图 9-14 所示。

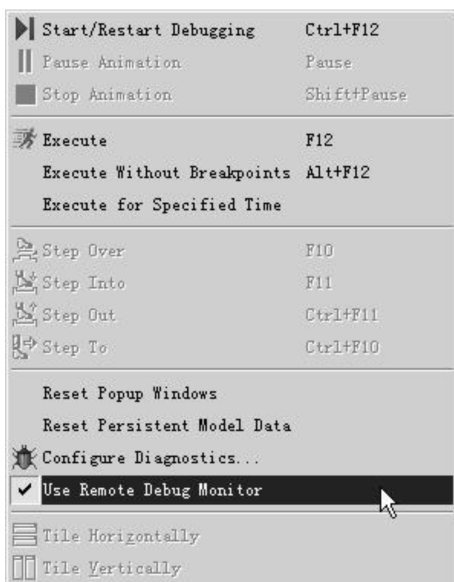


图 9-14 在【Debug】菜单中选择【Use Remote Debug Monitor】选项

- (5) 在 Keil 中编写 MCU 的程序。
- (6) 选择 Keil 的【Project】→【Options for Target' Target1'】, 如图 9-15 所示。

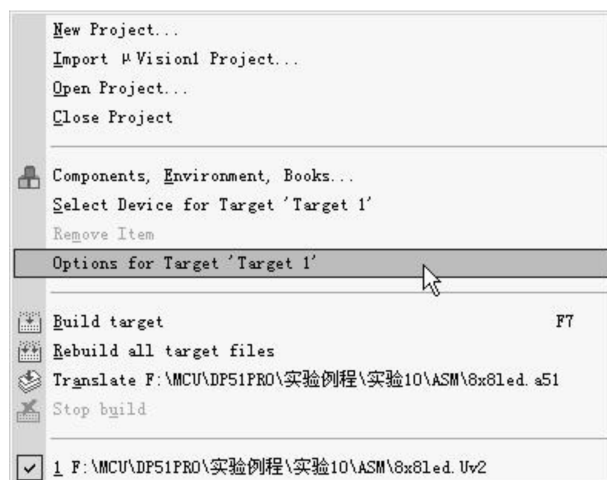


图 9-15 在【Project】菜单中选择【Options for Target' Target1'】

- (7) 在弹出的对话框中选择“Debug”选项卡，单击右上方的下三角按钮选择“Proteus VSM Monitor - 51 Driver”，如图 9-16 所示。

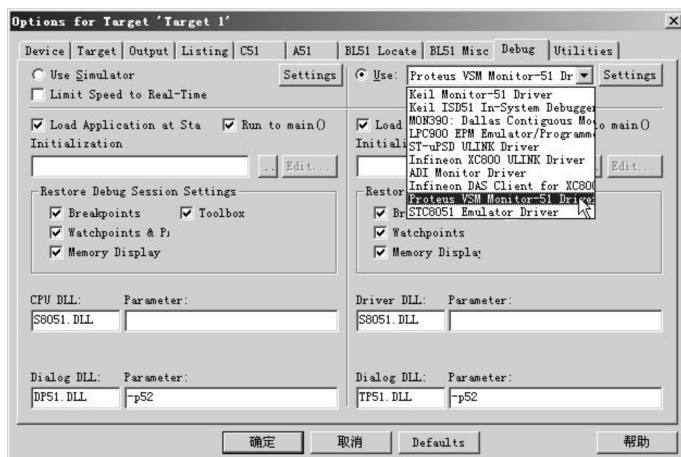


图 9-16 在“Debug”中选择“Proteus VSM Monitor - 51 Driver”

- (8) 单击“Settings”按钮，进入如图 9-17 所示对话框。

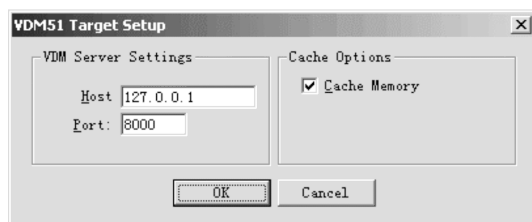


图 9-17 “VDM51 Target Setup”对话框

如果 Keil 程序与 PROTEUS 程序在同一台计算机上，IP 地址为 127.0.0.1；如果不在同一台计算机上，则填另一台计算机的 IP 地址，端口号一定为 8000。

(9) 在 Keil 中进行 Debug, 同时在 PROTEUS 中查看直观的结果。

## 2. PROTEUS ISIS 与 Keil C 的联合调试

**1) 硬件电路的设计、组装和调试** 硬件电路的设计包含两部分内容: 一是系统扩展, 即单片机自身的功能单元如 ROM、RAM、I/O 口、定时/计数器等容量不能满足应用系统的需要时, 必须在片外进行扩展, 选择适当的芯片, 设计相应的电路; 二是系统配置, 即按照系统的要求配置外围设备, 如键盘、显示器件、打印机、A/D 转换、D/A 转换等, 要设计合适的接口电路。

**2) 应用软件编制、调试程序** 使用 Keil 软件工具时, 项目开发流程和其他软件开发项目的流程极其相似。

(1) 创建一个项目, 从元器件库中选择目标元器件, 配置工具。

(2) 用汇编语言或 C51 语言创建源程序。

(3) 用项目管理器配置当前项目。

(4) 修改源程序中的错误。

(5) 测试, 连接应用。

**3) 总调** 总调是应用软件的链接调试, 程序固化, 软、硬件结合的应用系统调试。

总调就是软、硬件联合仿真系统由一个硬件执行环境和一个软件执行环境组成, 通常软件环境和硬件环境都有自己的除错和控制界面, Keil 与 PROTEUS 的整合调试可以实现系统总调, 在该系统中, Keil 作为软件调试界面, PROTEUS 作为硬件仿真和调试界面。



## 9.4 I/O 口的输入/输出应用

### 【任务】

读取 8 位双列直插开关状态, 并在 8 个发光二极管上显示其对应的状态。

### 【任务分析】

选取 P3 口读取 8 位双列直插开关的状态, 把结果送到 P0 口驱动对应的发光二极管。

### 【训练目的】

➤ 掌握 51 单片机 I/O 口输入的应用方法;

➤ 掌握 LED 的驱动方法。

## 1. PROTEUS ISIS 电路设计

**1) 元器件清单列表** 打开 PROTEUS ISIS 编程环境, 按照表 9-1 所列的清单添加元器件。

表 9-1 元器件清单

元器件名称	所属类	所属子类
AT89C52	Microprocessor ICs	8051 Family
DIPSW_8	Switches & Relays	Switches
LED-YELLOW	Optoelectronics	LEDs
MINRES330R	Resistors	0.6W Metal Film

2) 电路原理图 元器件全部添加后, 在 PROTEUS ISIS 的编辑区域中按图 9-18 所示原理图连接硬件电路。

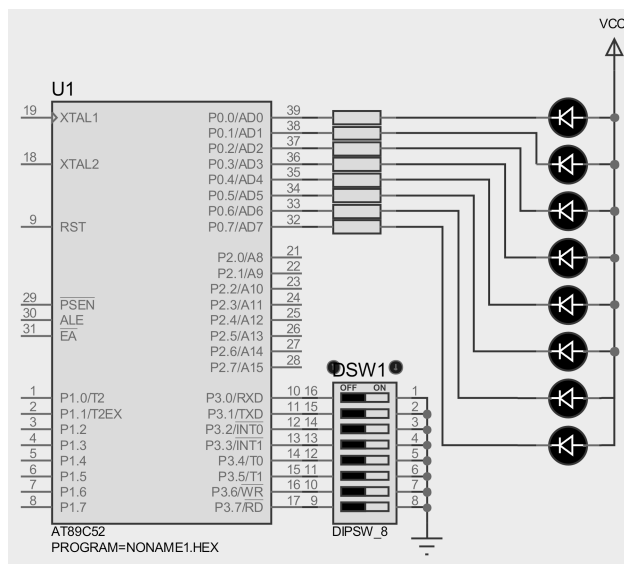


图 9-18 I/O 口电路原理图

## 2. 源程序设计

源程序如下:

```

ORG      0000H
AJMP     MAIN
ORG      0030H
MAIN:    MOV     SP, #60H
          MOV     P3, #0FFH
          MOV     A, P3
          MOV     P0, A
          LCALL    DELAY

          MOV     A, #0FFH
LOOP:    MOV     P0, A
          LCALL    DELAY
          DEC     A
          CJNE    A, #0FFH, LOOP
          AJMP     MAIN

DELAY:    MOV     R7, #01H
D1:      MOV     R6, #00H
D2:      MOV     R5, #00H
          DJNZ    R5, $
          DJNZ    R6, D2
          DJNZ    R7, D1
          RET
          END
  
```

### 3. PROTEUS ISIS 调试与仿真

参照 9.1 节建立程序文件，加载目标代码文件，进入调试环境执行程序。在 PROTEUS ISIS 界面中，运行仿真程序，8 位双列直插开关的状态会直接显示在 P0 口所驱动的 8 位发光二极管上，如图 9-19 所示。

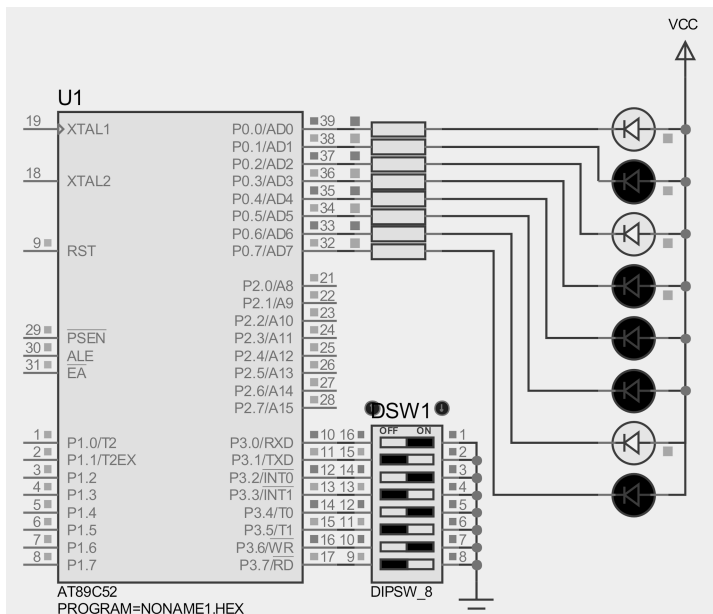


图 9-19 单片机 I/O 口仿真图

## 4. 小结

应用单片机 I/O 口驱动发光二极管时，需要注意其驱动能力。



## 9.5 按键识别

### 【任务】

用单片机 P2 端口设计 16 只按键的识别电路，并把对应的按键值 0 ~ F 用数码管显示出来。

### 【任务分析】

P2 口识别 16 只按键, 需要进行行列式设计, 4 行  $\times$  4 列共 16 只按键, P2 刚好具有 8 个 I/O 口, 满足要求。显示 0 ~ F 一片 7 段数码管就可以, 选择用 P3 口驱动共阴极数码管。

### 【训练目的】

➤ 掌握按键扫描及处理程序的编程方法和调试方法。

## 1. PROTEUS ISIS 电路设计

1) 元器件清单列表 打开 PROTEUS ISIS 编程环境, 按照表 9-2 所列的清单添加元器件。

表 9-2 元器件清单

元器件名称	所 属 类	所 属 子 类
AT89C51	Microprocessor ICs	8051 Family
7SEG-COM-CAT-BLUE	Optoelectronics	7-Segment Displays
BUTTON	Switches & Relays	Switches
CHIPRES10K	Resistors	Chip Resistor

2) 电路原理图 元器件全部添加后, 在 PROTEUS ISIS 的编辑区域中按图 9-20 所示原理图连接硬件电路。

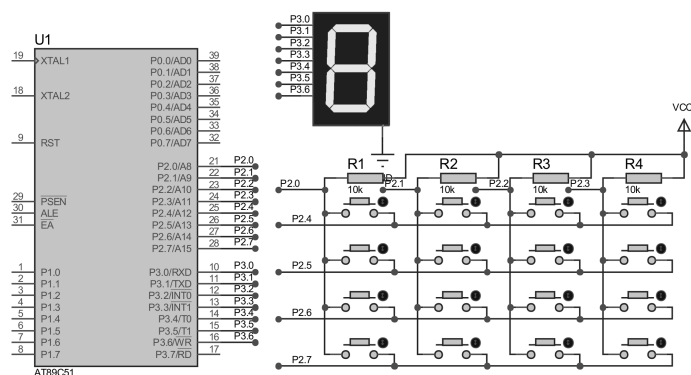


图 9-20 电路原理图

## 2. 源程序设计

源程序如下:

```

                ORG      0000H
                AJMP     MAIN
                ORG      0030H
MAIN:           MOV      SP, #60H
                MOV      A, #10H
                LCALL    DISPLAY

LOOP:           LCALL    JUMP_BUTTON
                LCALL    DISPLAY
                AJMP     LOOP

JUMP_BUTTON:   MOV      A, #10H
                LCALL    SCAN_BUTTON
                XRL      A, 10H
                JZ        JUMP_BUTTON
                MOV      A, 10H
                LCALL    DEL10MS
                LCALL    SCAN_BUTTON
                XRL      A, 10H
                JNZ       JUMP_BUTTON
                MOV      A, 10H

```



```

RET

SCAN_BUTTON:
    CLR    P2.4
    JB     P2.0, YI
    MOV    10H, #00
    AJMP   JIESHU
YI:        JB     P2.1, ER
    MOV    10H, #01
    AJMP   JIESHU
ER:        JB     P2.2, SAN
    MOV    10H, #02H
    AJMP   JIESHU
SAN:       JB     P2.3, SI
    MOV    10H, #03H
    AJMP   JIESHU
SI:        CLR    P2.5
    JB     P2.0, WU
    MOV    10H, #04H
    AJMP   JIESHU
WU:        JB     P2.1, LIU
    MOV    10H, #05H
    AJMP   JIESHU
LIU:       JB     P2.2, QI
    MOV    10H, #06H
    AJMP   JIESHU
QI:        JB     P2.3, BA
    MOV    10H, #07H
    AJMP   JIESHU
BA:        CLR    P2.6
    JB     P2.0, JIU
    MOV    10H, #08H
    AJMP   JIESHU
JIU:       JB     P2.1, SHI
    MOV    10H, #09
    AJMP   JIESHU
SHI:       JB     P2.2, SHIYI
    MOV    10H, #0AH
    AJMP   JIESHU
SHIYI:     JB     P2.3, SHIER
    MOV    10H, #0BH
    AJMP   JIESHU
SHIER:     CLR    P2.7
    JB     P2.0, SHISAN
    MOV    10H, #0CH
    AJMP   JIESHU
SHISAN:    JB     P2.1, SHISI
    MOV    10H, #0DH
    AJMP   JIESHU
SHISI:     JB     P2.2, SHIWU

```

```

MOV     10H, #0EH
AJMP    JIESHU
SHIWU:  JB     P2.3, SHILIU
MOV     10H, #0FH
AJMP    JIESHU
SHILIU: MOV     10H, #10H
JIESHU: SETB    P2.4
        SETB    P2.5
        SETB    P2.6
        SETB    P2.7
        RET

DEL10MS: MOV R5, #0F4H
DL10MS0: MOV R6, #013H
        DJNZ R6, $
        DJNZ R5, DL10MS0
        RET

DISPLAY:
MOV     DPTR, #TAB
MOVC    A, @ A + DPTR
MOV     P3, A
        RET

TAB:    DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
        DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
        DB 00H
        END

```

### 3. PROTEUS ISIS 调试与仿真

参照 9.1 节建立程序文件，加载目标代码文件，进入调试环境执行程序。在 PROTEUS ISIS 界面中，任意按下一个按键，在数码管中会显示对应的键值，如图 9-21 所示。

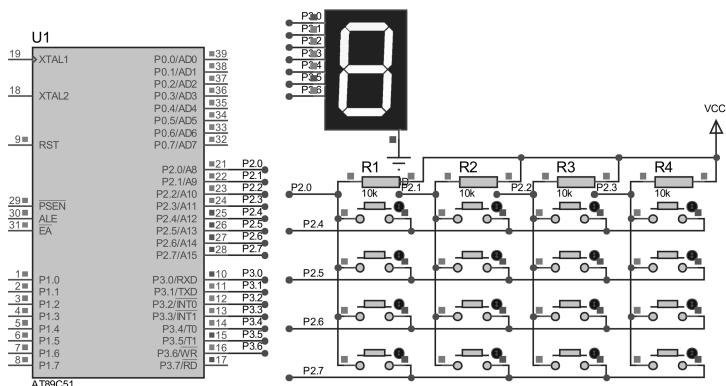


图 9-21 按键识别仿真图

### 4. 小结

- 机械式按键的扫描需要硬件消抖和软件消抖配合使用。
- 行列式按键相对于独立按键的确能够节约很多硬件资源。在一些对硬件资源要求比较多的情况下，可以选择如 ZLG7289 等按键扫描的专用芯片。



## 9.6 数码管显示

### 【任务】

用数码管循环显示 0 ~ F。

### 【任务分析】

循环显示 0 ~ F，选用一个 7 段数码管就可以。只是数码管需要驱动，可选择 74HC595 来驱动数码管，以减少系统显示硬件的开支。

### 【训练目的】

- 掌握单片机驱动 7 段数码管显示数字的编程方法。
- 掌握串入并出集成芯片 74LS595 的使用方法。

### 1. PROTEUS ISIS 电路设计

**1) 元器件清单列表** 打开 PROTEUS ISIS 编程环境，按照表 9-3 所列的清单添加元器件。

表 9-3 元器件清单

元器件名称	所属类	所属子类
AT89C51	Microprocessor ICs	8051 Family
74HC595	TTL 74HC series	Registers
7SEG-MPX1-CC	Optoelectronics	7-Segment Displays
MINRES300R	Resistors	0.6W Metal Film

**2) 电路原理图** 元器件全部添加后，在 PROTEUS ISIS 的编辑区域中按图 9-22 所示原理图连接硬件电路。

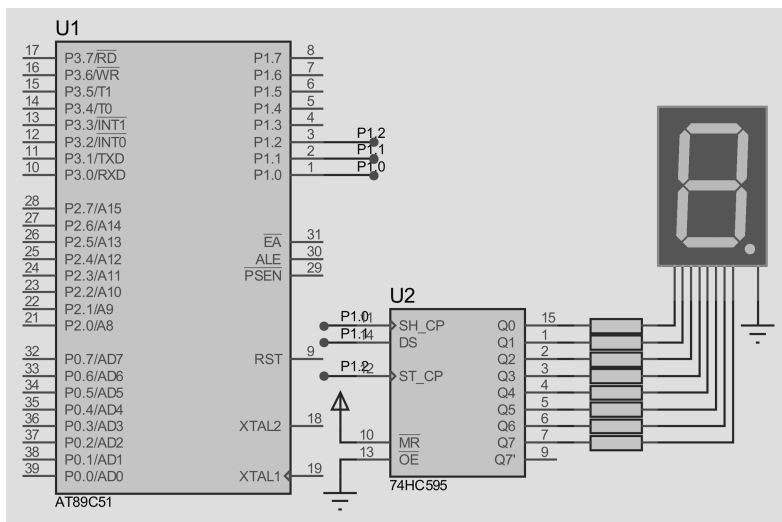


图 9-22 电路原理图

## 2. 源程序设计

源程序如下：

```
ORG      0000H
        AJMP      MAIN
        ORG      0030H
MAIN:    MOV      SP,#60H
        MOV      30H,#10H
        MOV      A,30H
        LCALL     DISPLAY
LOOP:    MOV      30H,#00H
LOOP1:   MOV      A,30H
        LCALL     DISPLAY
        LCALL     DEL1S
        INC      30H
        MOV      R1,30H
        CJNE     R1,#10H,LOOP1
        AJMP     LOOP

DISPLAY: MOV      DPTR,#TAB
        MOVC     A,@ A + DPTR
        MOV      R4,#08H
TRANS_LOOP:
        RLC      A
        MOV      P1.1, C
        CLR      P1.0
        NOP
        NOP
        SETB     P1.0
        NOP
        CLR      P1.2
        NOP
        NOP
        SETB     P1.2
        NOP
        NOP
        CLR      P1.2
        DJNZ     R4, TRANS_LOOP
        RET

TAB:     DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
        DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
        DB 00H

DEL1S:   MOV      R5, #0FEH
DL1S0:   MOV      R6, #0E7H
```

```

DL1S1:  MOV     R7, #07H
        DJNZ    R7, $
        DJNZ    R6, DL1S1
        DJNZ    R5, DL1S0
        RET
        END

```

### 3. PROTEUS ISIS 调试与仿真

参照 9.1 节建立程序文件，加载目标代码文件，进入调试环境执行程序。在 PROTEUS ISIS 界面中，运行程序后，会在数码管上循环显示 0 ~ F，如图 9-23 所示。

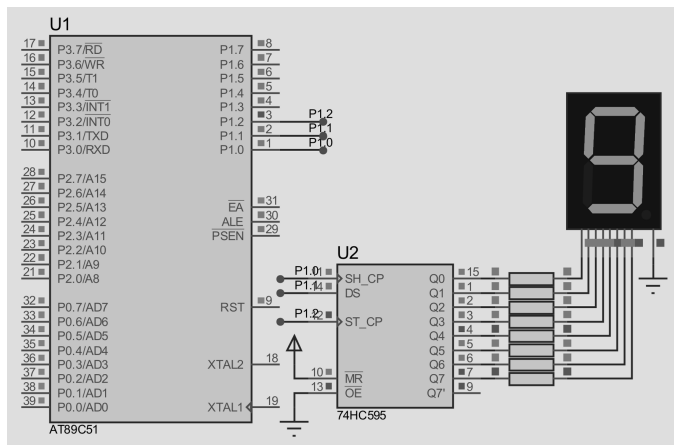


图 9-23 数码管显示仿真图

### 4. 小结

- 数码管驱动还可以选择 74HC164 代替。
- 在数码管应用较多的场合，尽量选用动态数码管，以节约 I/O 口的开支。



## 9.7 外部中断实验

### 【任务】

应用下降沿中断响应或低电平中断响应编写中断程序，并能检验中断优先级的相应原则。

### 【任务分析】

首先能够提供中断产生的低电平或下降沿，选用按键来完成。能够验证中断优先级的原则，涉及两个方面：一个是主程序被中断所中断，另一个是需要两个中断源且处理不同的任务。选择两个外部中断，每个中断的任务是在不同的数码管上显示数字。

### 【训练目的】

- 学习外部中断的设置与中断子程序的编写并掌握中断优先级的原则。

## 1. PROTEUS ISIS 电路设计

1) 元器件清单列表 打开 PROTEUS ISIS 编程环境, 按照表 9-4 所列的清单添加元器件。

表 9-4 元器件清单

元器件名称	所属类	所属子类
AT89C51	Microprocessor ICs	8051 Family
BUTTON	Switches & Relsys	Switches
7SEG-COM-ANODE	Optoelectronics	7-Segment Displays
MINRES300R	Resistors	0. 6W Metal Film

2) 电路原理图 元器件全部添加后, 在 PROTEUS ISIS 的编辑区域中按图 9-24 所示原理图连接硬件电路。

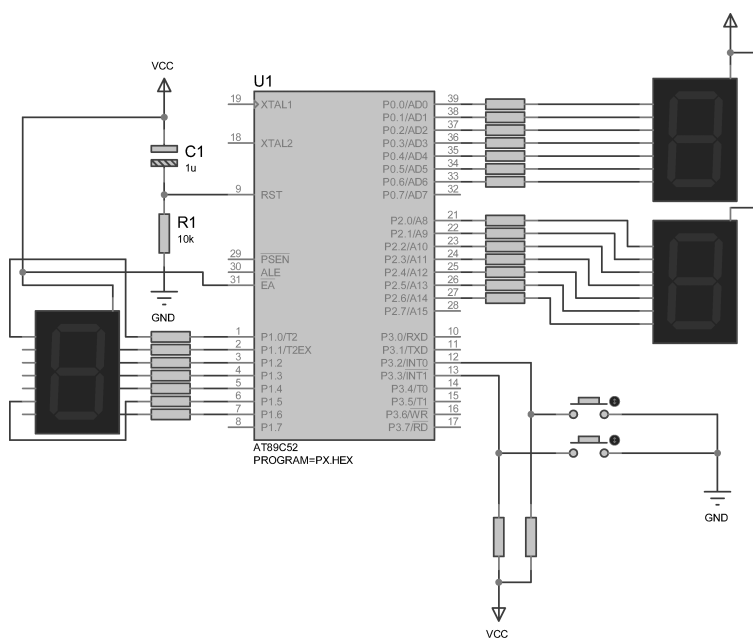


图 9-24 电路原理图

## 2. 源程序设计

源程序如下:

```

ORG      0000H
AJMP     MAIN
ORG      0003H
AJMP     EX0_INT
ORG      0013H
AJMP     EX1_INT
ORG      0030H
MAIN:    MOV     SP, #60H
CLR      P1.3
SETB     EA
SETB     IT0

```

```

SETB    IT1
SETB    EX0
SETB    EX1
SETB    PX0
CLR     PX1

MOV     30H, #11H
MOV     31H, #11H
MOV     32H, #11H
MOV     33H, #11H
LCALL   DISPLAY
LCALL   DEL1S
LCALL   DEL1S

LOOP:
MOV     R1, #00H
LOOP1:  MOV     A, R1
        LCALL   TRANS
        LCALL   DEL1S
        INC     R1
        CJNE    R1, #10H, LOOP1
        AJMP    LOOP

EX0_INT:MOV     A, #0FEH
        MOV     P2, A
        LCALL   DEL1S
        RL      A
        MOV     P2, A
        LCALL   DEL1S
        RL      A
        MOV     P2, A
        LCALL   DEL1S
        RL      A
        MOV     P2, A
        LCALL   DEL1S
        RL      A
        MOV     P2, A
        LCALL   DEL1S
        RL      A
        MOV     P2, A
        LCALL   DEL1S
        MOV     A, #10101010
        MOV     P2, A
        LCALL   DEL1S
        CPL     A
        MOV     P2, A
        LCALL   DEL1S
        CPL     A
        MOV     P2, A
        LCALL   DEL1S
        MOV     P2, #0FFH

```

```

        RETI

EX1_INT:MOV    A, #0FEH
        MOV     P0, A
        LCALL   DEL1S
        RL      A
        MOV     P0, A
        LCALL   DEL1S
        RL      A
        MOV     P0, A
        LCALL   DEL1S
        RL      A
        MOV     P0, A
        LCALL   DEL1S
        RL      A
        MOV     P0, A
        LCALL   DEL1S
        RL      A
        MOV     P0, A
        LCALL   DEL1S
        MOV     A, #10101010
        MOV     P0, A
        LCALL   DEL1S
        CPL     A
        MOV     P0, A
        LCALL   DEL1S
        CPL     A
        MOV     P0, A
        LCALL   DEL1S
        MOV     P0, #0FFH
        RETI

DISPLAY:
        MOV     R0, #30H
DISP_LOOP:
        MOV     A, @ R0
        LCALL   TRANS
        INC     R0
        CJNE    R0, #34H, DISP_LOOP
        RET

TRANS:  MOV     DPTR, #TAB
        MOVC    A, @ A + DPTR
        MOV     R4, #08H
TRANS_LOOP:
        RRC     A
        MOV     P1.6, C
        CLR     P1.7
        SETB    P1.7
        DJNZ    R4, TRANS_LOOP
        RET

```



```

TAB:      DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
          DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
          DB 00H, 40H

DEL1S:    MOV     R5, #0FEH
DL1S0:    MOV     R6, #0E7H
DL1S1:    MOV     R7, #07H
          DJNZ    R7, $
          DJNZ    R6, DL1S1
          DJNZ    R5, DL1S0
          RET

          END

```

### 3. PROTEUS ISIS 调试与仿真

参照 9.1 节建立程序文件，加载目标代码文件，进入调试环境执行程序。在 PROTEUS ISIS 界面中，分别按下两个按键，观察主程序被外部中断所中断的情况。在一个中断执行的过程中，按下另外一个按键，观察有没有中断嵌套，低优先级中断有没有被高优先级中断所中断，高优先级中断响应的过程中是否会被低优先级中断所中断。部分结果如图 9-25 所示。

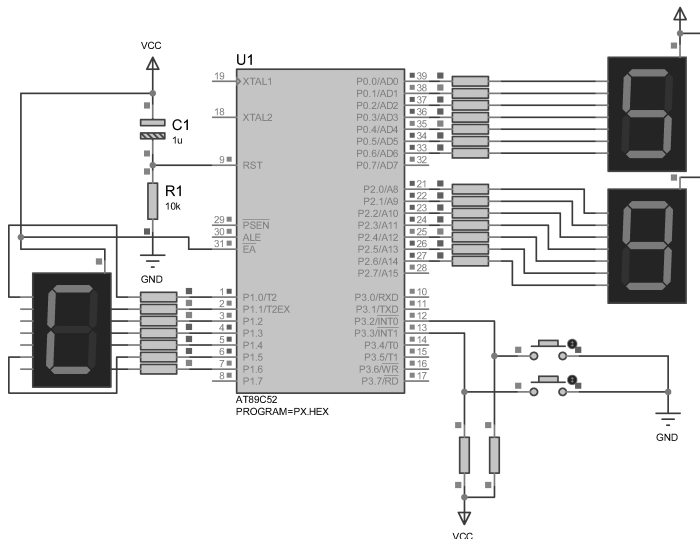


图 9-25 中断优先级验证仿真图

### 4. 小结

- 涉及中断嵌套，在每个中断里需要对各寄存器进行保护。
- 多个中断同时触发时，把两个外部中断的输入信号产生按键换成一个。



## 9.8 定时/计数器应用

### 【任务】

设计一个单片机应用系统，用于足球比赛中控球率的统计。

【任务分析】

统计控球率，首先需要分别统计主、客队的控球时间，每个队的控球时间与总的比赛进行时间之比就是该队的控球率。需要解决的根本问题是统计时间。

【训练目的】

➤ 掌握 AT89C51 单片机定时/计数器及外部中断的使用方法。

1. PROTEUS ISIS 电路设计

记录主、客队的控球时间，应用单片机的两个定时器进行定时。时间显示应用两个 4 位共阴极数码管进行，数码管的驱动选择三极管驱动。设计两个按键用于主、客队控球计时启动，一个按键用于暂停使用。

1) 元器件清单列表 打开 PROTEUS ISIS 编辑环境，按表 9-5 所列的清单添加元件。

表 9-5 元器件清单

元器件名称	所属类	所属子类
AT89C51	Microprocessor ICs	8051 Family
CAP	Capacitors	Generic
CAP-ELEC	Capacitors	Generic
CRYSTAL	Miscellaneous	—
RES	Resistors	Generic
BUTTON	Switches & Relays	Switches
74HC04	TTL 74HC series	Gates & Inverters
FZT658	Transistors	Bipolar
7SEG-MPX4-CA	Optoelectronics	7 - Segment Displays

2) 电路原理图 元件全部添加完成后，在 PROTEUS ISIS 的编辑区域中按图 9-26 所示电路原理图连接硬件电路。

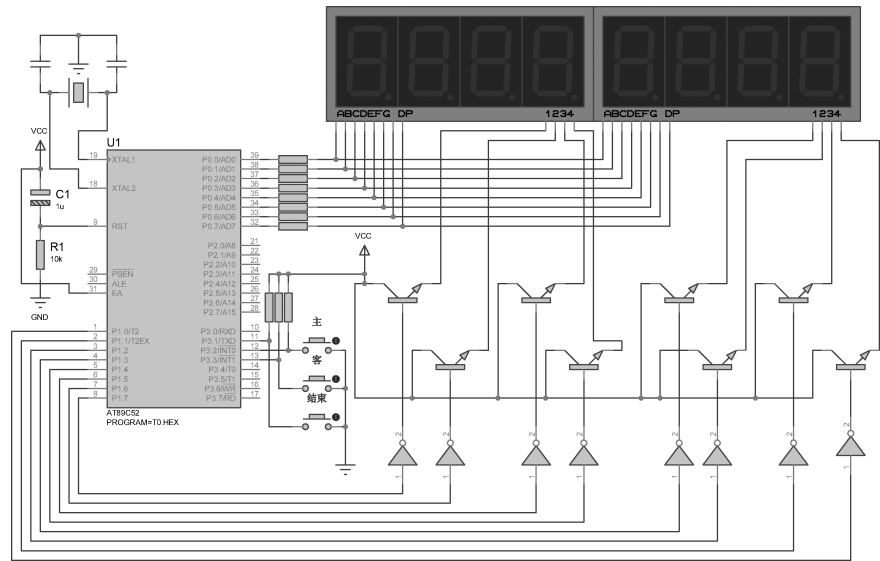


图 9-26 电路原理图

## 2. 源程序设计

源程序设计由初始化程序、主程序、中断子程序及其他子程序 4 部分组成。其中中断子程序又由 4 个中断服务子程序组成，分别为定时器 T0、定时器 T1、外部中断 0、外部中断 1 中断服务子程序。

定时器 T0 和 T1 定时时长为 50ms，每 20 次也就是定时 1s 计时值更新一次，所以取 2EH 和 2FH 两个字节用来存取定时的次数，每 20 次一个循环。

按下主队按键，记录主队控球时间；按下客队按键，主队计时停止，客队控球时间开始记录，如此反复。按下暂停按键，主、客队计时停止。再次按下主队或客队按键，重新开始计时。

选择 RAM 中的 30H、31H、32H、33H 用于记录主队控球时间，其中 30H 和 31H 记录秒，30H 应用十进制，31H 应用六进制，32H 和 33H 记录分，32H 和 33H 应用十进制。选择 34H、35H、36H、37H 用于记录客队控球时间。

**1) 初始化程序** 初始化程序主要完成对 2EH、2FH 及用于存储主、客对控球时长的 8 字节（30H ~ 37H）赋初值 0，定时器的工作方式设定、定时初值装入、外部中断的触发方式设置、中断优先级设置、外部中断及定时器中断使能设置等。初始化程序比较简单，流程图略。初始化详细程序如下：

```

MOV      A, #00H
MOV      R0, #2EH
LOOP1:  MOV      @R0, A
        INC      R0
        CJNE     R0, #38H, LOOP1
MOV      TMOD, #11H
MOV      TL0, #0B0H
MOV      TL1, #0B0H
MOV      TH0, #3CH
MOV      TH1, #3CH
SETB     ET0
SETB     ET1
SETB     EX0
SETB     EX1
SETB     PT0
SETB     PT1
SETB     EA

```

**2) 主程序** 主程序的功能比较单一，就是完成对暂停按键的判断。如果检测到暂停按键按下，就暂停定时器 T0 和 T1 的定时功能。主程序比较简单，流程图略。主程序详细程序如下：

```

M_FUN:  JB      P3.1, GO
        CLR      TR0
        CLR      TR1
GO:
        AJMP     M_FUN

```

## 3) 中断服务子程序

(1) 定时器 T0 中断服务子程序。定时器 T0 中断服务子程序的主要功能是定时，用于记

录主队的控球时间。定时每满 1s，更新计时值，并显示一次。由于使用的是工作模式 1，每次进入中断需要重新装入定时器的计数初值。定时器 T0 中断服务子程序的流程图如图 9-27 所示。

定时器 T0 中断服务详细子程序如下：

```

T0_S:  MOV     TH0, #3CH
        MOV     TL0, #0B0H
        INC     2EH
        MOV     R7, 2EH
        CJNE    R7, #20, NEXT0
        MOV     2EH, #00H
        INC     30H
        MOV     R7, 30H
        CJNE    R7, #10, NEXT
        INC     31H
        MOV     30H, #00H
        MOV     R7, 31H
        CJNE    R7, #6, NEXT
        MOV     31H, #00H
        INC     32H
        MOV     R7, 32H
        CJNE    R7, #10, NEXT
        MOV     32H, #00H
        INC     33H
        MOV     R7, 33H
        CJNE    R7, #10, NEXT
        MOV     33H, #00H
NEXT:   LCALL   DISPLAY
NEXT0:  RETI

```

(2) 定时器 T1 中断服务子程序。定时器 T1 中断服务子程序的主要功能是定时，用于记录客队的控球时间。定时每满 1s，更新客队计时值，并显示一次。同样由于使用的是工作模式 1，每次进入中断需要重新装入定时器的计数初值。定时器 T1 中断服务子程序的流程图如图 9-28 所示。

定时器 T1 中断服务详细子程序如下：

```

T1_S:  MOV     TH1, #3CH
        MOV     TL1, #0B0H
        INC     2FH
        MOV     R4, 2FH
        CJNE    R4, #20, NEXT3
        MOV     2FH, #00H
        INC     34H
        MOV     R4, 34H
        CJNE    R4, #10, NEXT2
        INC     35H
        MOV     34H, #00H
        MOV     R4, 35H
        CJNE    R4, #6, NEXT2

```

```

MOV     35H, #00H
INC     36H
MOV     R4, 36H
CJNE    R4, #10, NEXT2
MOV     36H, #00H
INC     37H
MOV     R4, 37H
CJNE    R4, #10, NEXT2
MOV     37H, #00H
NEXT3:  LCALL  DISPLAY
NEXT2:  RETI

```

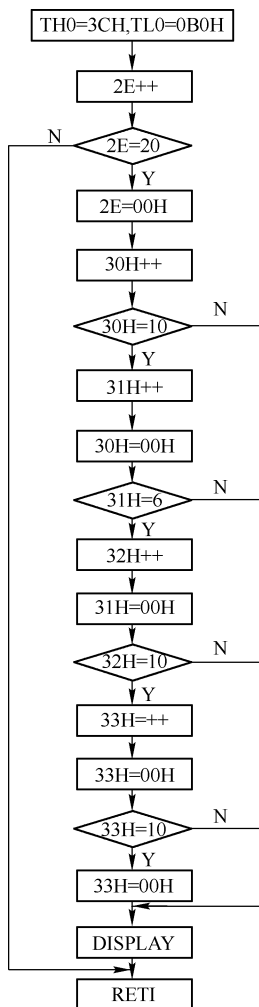


图 9-27 定时器 T0 中断服务子程序流程图

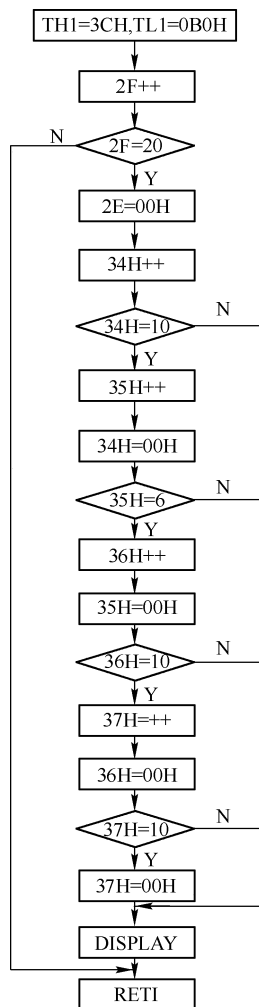


图 9-28 定时器 T1 中断服务子程序流程图

(3) 外部中断 0 中断服务子程序。外部中断 0 中断服务子程序的功能，只需完成关闭定时器 T1，启动定时器 T0 即可。流程图如图 9-29 所示。

外部中断 0 中断服务详细子程序如下：

```

INT0_S: CLR     TR1
        SETB    TR0

```

RETI

(4) 外部中断 1 中断服务子程序。外部中断 1 中断服务子程序的功能与外部中断 0 中断服务子程序的功能相反, 只需完成关闭定时器 T0, 启动定时器 T1 即可。流程图如图 9-30 所示。

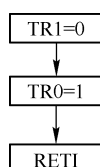


图 9-29 外部中断 0 中断服务子程序流程图

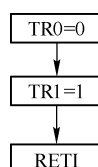


图 9-30 外部中断 1 中断服务子程序流程图

外部中断 1 中断服务详细子程序如下:

```

INT1_S: CLR    TR0
        SETB   TR1
        RETI
  
```

**4) 其他子程序** 其他子程序包括延时子程序和显示子程序两部分, 延时子程序用于数码管动态显示时的延时, 显示子程序用于把记录的时间发送到数码管上显示。

(1) 延时子程序:

```

DEL10MS: MOV    R5, #0EEH
DL5MS0:  MOV    R6, #09H
        DJNZ    R6, $
        DJNZ    R5, DL5MS0
        RET
  
```

(2) 显示子程序:

```

DISPLAY:
        MOV     DPTR, #TAB
NEXT1:
        MOV     R0, #30H
        MOV     R3, #7FH
LOOP:
        MOV     A, @ R0
        MOVC    A, @ A + DPTR
        MOV     P0, A
        MOV     A, R3
        RL      A
        MOV     P1, A
        MOV     R3, A
        LCALL   DEL10MS
        MOV     P1, #0FFH
        INC     R0
        CJNE    R0, #38H, LOOP
        JB      P3.1, GO
        CLR     TR0
        CLR     TR1
  
```

GO:

```
AJMP      NEXT1
```

RET

共阴极数码管显示 0 ~ F 码表:

TAB:DB0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H

### 3. PROTEUS ISIS 调试与仿真

参照 9.1 节建立程序文件，加载目标代码文件，进入调试环境执行程序。在 PROTEUS ISIS 界面中，交替按下主、客队按键进行时间记录，结束时，按下结束按钮，可以看到数码管显示的时间变化，如图 9-31 所示。

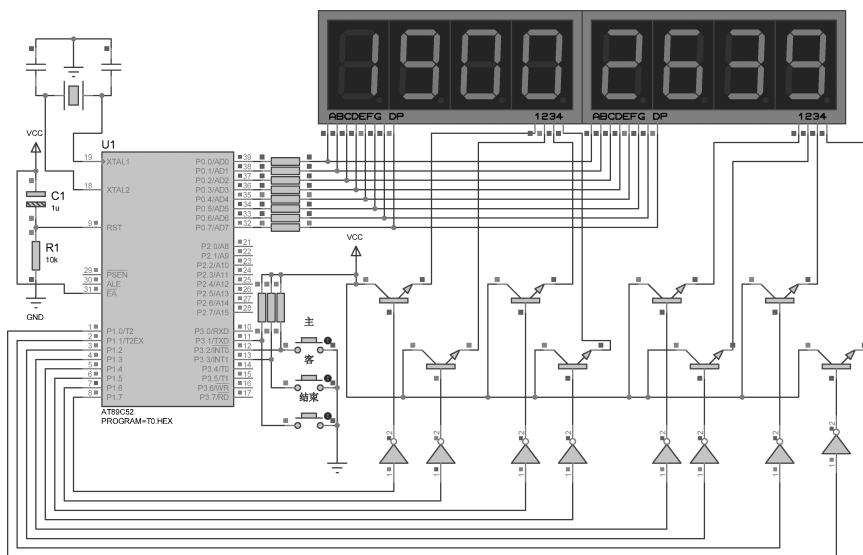


图 9-31 程序运行结果

## 4. 小结

- AT89C51 单片机的应用过程中如何有多个中断源, 需要根据实际情况设置各个中断源的中断优先级; 同级的多个中断源同时申请中断时, 按照外部中断 0、定时器 T0、外部中断 1、定时器 T1、串行口中断由高到低的自然优先级顺序响应中断。
- 在 PROTEUS 与 Keil 联合调试过程中, 可以综合运用 Keil 中的多种调试功能来仔细观察电路的工作情况。



## 9.9 思考与练习

1. 设计 4 位动态数码管显示电路，并编程进行测试。
2. 假设单片机晶振频率为 6MHz，要求由 P1.0 端口输出一个周期为 500 $\mu$ s 的方波信号，由定时器 T1 采用查询方法实现。

3. 假设单片机晶振频率为 6MHz，使用定时器 T1 的定时模式 1，在 P1.0 端口输出周期为  $20\mu\text{s}$ 、占空比（高电平时间占整个时间的比例）为 60% 的矩形脉冲，以查询方式编写程序。

4. 用 4 个指示灯对应显示 4 个开关的状态，即开关打开时对应指示灯灭，开关闭合时对应指示灯亮；由另外一个开关控制中断申请，要求每申请一次中断，完成一个读/写过程，使指示灯与开关状态对应。

5. 在键盘设计工程中，考虑如何结合硬件和软件对按键进行消抖处理。

6. 用 AT89C51 串行口外接 74HC164 串入并出，为寄存器扩展 8 位并行口。8 位并行口的每一位都接一个发光二极管，要求发光二极管从左到右以一定延时轮流显示，并不断循环。

7. 用 AT89C51 串行口外加移位寄存器 74HC165 扩展 8 位输入口，输入数据由 8 个开关提供，另有一个开关 K 提供联络信号。设置一个按键作控制用，输入的 8 位为开关量，提供逻辑模拟子程序的输入信号。



# 职 场 篇

# 第 10 章 低频信号发生器的设计与仿真

低频信号发生器采用单片机波形合成发生器产生高精度、低失真的正弦波电压，可用于校验频率继电器、同步继电器等，也可作为低频变频电源使用。



## 10.1 系统功能

本章以单片机为核心设计一个低频函数信号发生器。信号发生器采用数字波形合成技术，通过硬件电路和软件程序相结合，可输出自定义波形，如正弦波、方波、三角波及其他任意波形。波形的频率和幅度在一定范围内可任意改变。本章除介绍波形的生成原理、硬件电路和软件部分的设计原理以外，还介绍单片机控制 D/A 转换器产生上述信号的硬件电路和软件编程、DAC0832 D/A 转换器的原理和使用方法、AT89C52，以及与设计电路有关的各种芯片、关于产生不同低频信号的信号源的设计方案。该低频信号发生器具有体积小、价格低、性能稳定、功能齐全的优点。



## 10.2 设计方案

系统采用一片 AT89C52 单片机作为数据输出，经 DAC0832 输出模拟量，经过 OP07 信号放大，输入到示波器。单片机 P2 口作为数据输出端口，P1.0、P1.1、P1.2 外接按键，分别作为频率增加、频率减小、波形选择控制键。

DAC0832 是 8 位分辨率的并行 D/A 转换集成芯片，采用 R-2R 倒梯形电阻网络，与 TTL 电平完全兼容，输出电流建立时间仅为  $1\mu\text{s}$ ，功率消耗为  $20\text{mW}$ ，电源供电为  $5 \sim 15\text{V}$ ，其以接口简单、价格低廉、转换容易控制等优点，在单片机系统中广泛应用。图 10-1 为 DAC0832 的内部逻辑图。

DAC0832 的引脚说明如表 10-1 所示，引脚如图 10-2 所示。

当  $\overline{\text{ILE}}$ 、 $\overline{\text{CS}}$ 、 $\overline{\text{WR1}}$  信号同时有效时，输入寄存器的输出随输入变化；当  $\overline{\text{WR2}}$ 、 $\overline{\text{XFER}}$  信号同时有效时，DAC 寄存器的输出随输入变化。

电流输出端电流计算公式如下：

$$I_{\text{out1}} = \frac{V_{\text{REF}}}{15\text{k}\Omega} \times \frac{\text{Digital input}}{256}$$
$$I_{\text{out2}} = \frac{V_{\text{REF}}}{15\text{k}\Omega} \times \frac{255 - \text{Digital input}}{256}$$

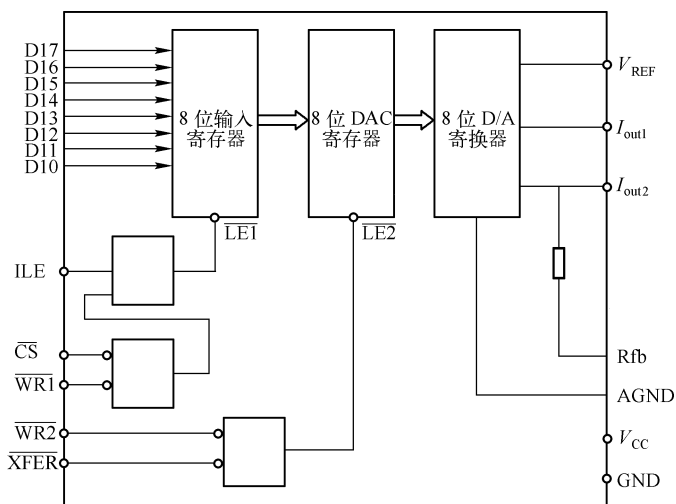


图 10-1 DAC0832 内部逻辑图

表 10-1 DAC0832 引脚说明

引 脚	符 号	说 明
1	$\overline{\text{CS}}$	片选信号，低电平有效
2	$\overline{\text{WR1}}$	输入寄存器写选通信号
3	GND	电源地
4~7	D3~D0	输数据输入端低 4 位，TTL 电平
8	$V_{\text{REF}}$	基准电压输入，在 $-10 \sim 10\text{V}$ 之间
9	Rfb	反馈信号输入脚，内部有反馈电阻
10	GND	电源地
11	$I_{\text{out1}}$	输出电流信号，当输入全为 1 时，电流输出最大
12	$I_{\text{out2}}$	互补电流信号输出端，与 $I_{\text{out1}}$ 的和为常数
13~16	D7~D4	数据输入端高 4 位，TTL 电平
17	$\overline{\text{XFER}}$	数据传送信号控制输入端，低电平有效
18	$\overline{\text{WR2}}$	DAC 寄存器写选通信号
19	ILE	输入锁存信号，高电平有效
20	$V_{\text{CC}}$	电源输入端，在 $+5 \sim +15\text{V}$ 之间

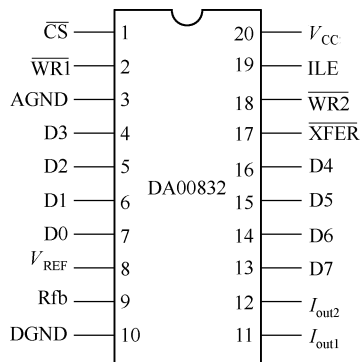


图 10-2 DAC0832 引脚图

DAC0832 的工作方式如下。

1) 单缓冲工作方式 此时一个寄存器处于直通状态，另一个处于受控状态。在不要求多相 D/A 同时输出时，可采用单缓冲方式，只需一次写操作便可进行转换，可大大提高数据的吞吐量。

2) 双缓冲工作方式 此时两个寄存器均处于受控状态。当要求多个模拟量同时输出时，可采用双缓冲工作方式。



## 10.3 系统硬件仿真电路

在元件库中提取元器件，并放入界面中。所用元器件如表 10-2 所示。

表 10-2 元器件列表

元器件名称	类 别	子 类	标 识
AT89C52	MCS8051	—	U1
RES	DEVICE	Generic	R
BUTTON	ACTIVE	—	—
DAC0832	NATDAC	—	U2

连接元器件，并进行电气检测。电路原理图如图 10-3 所示。

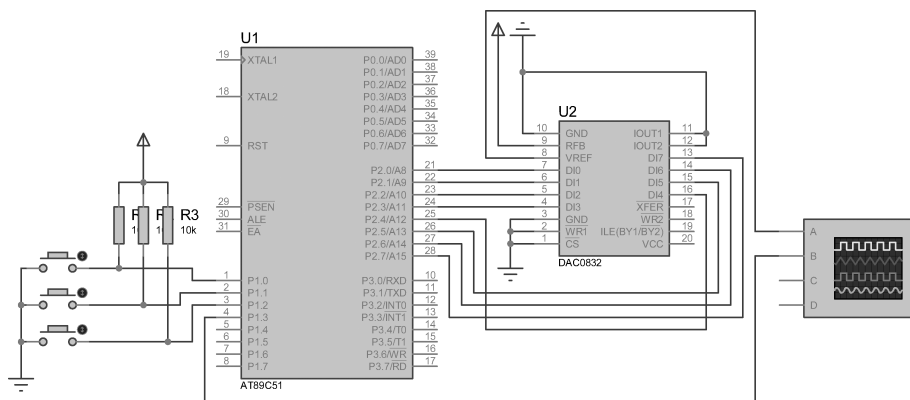


图 10-3 电路原理图



## 10.4 程序设计

### 1. 初始化程序

初始化程序的主要工作是设置定时器的工作模式、初值预置、开中断、打开定时器等。此处定时器 T0 工作于 16 位定时模式，单片机按定时时间重复地把波形数据送到 DAC0832 的寄存器。初始化程序流程图如图 10-4 所示。

### 2. 按键扫描程序

按键扫描程序的任务是检查 3 个按键是否被按下，如果有按键按下则执行相应的功能。这里 3 个按键分别用于波形频率增加、波形频率减小和波形选择，其程序流程图如图 10-5 所示。

### 3. 波形数据产生程序

波形数据产生程序是定时器 T0 的中断程序，当定时器计数溢出时发生一次中断。当发生中断时，单片机将按次序将波形数据表中的波形数据送入 DAC0832，DAC0832 根据输入

的数据大小输出相应的电压。波形数据产生程序流程图如图 10-6 所示。

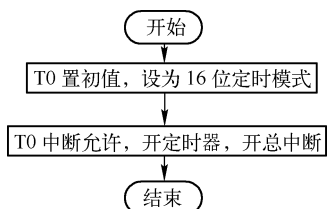


图 10-4 初始化程序流程图

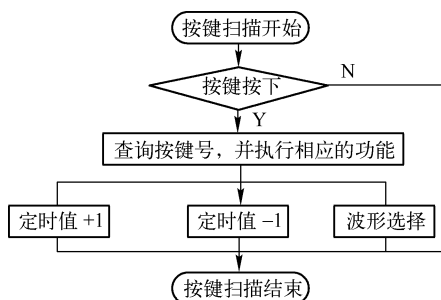


图 10-5 按键扫描程序流程图

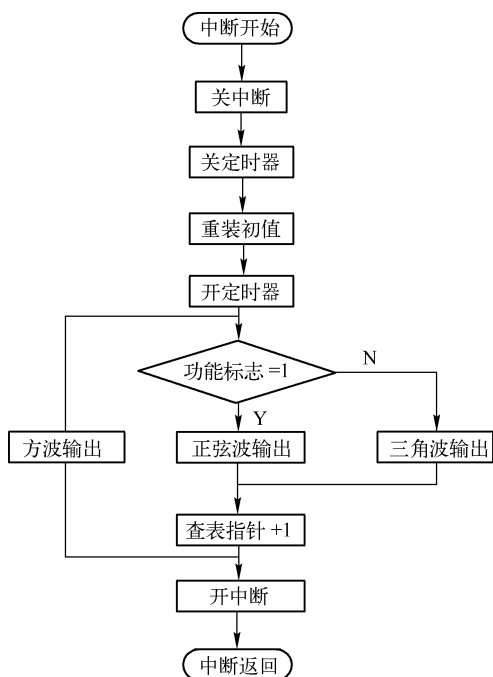


图 10-6 波形数据产生程序流程图



## 10.5 软件调试与运行结果

程序设计完成后, 将程序产生的 “.HEX” 文件导入 PROTEUS ISIS 仿真电路的 AT89C52 中, 单击仿真按钮, 开始仿真。仿真结果如图 10-7 ~ 图 10-9 所示。

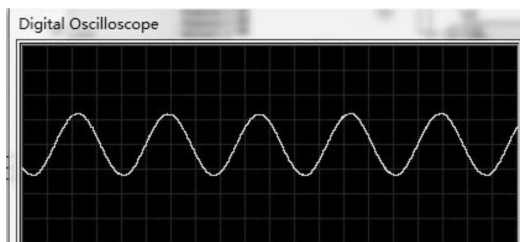


图 10-7 正弦波

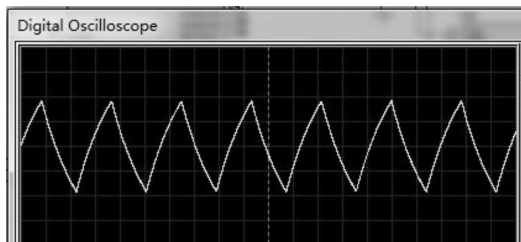


图 10-8 锯齿波

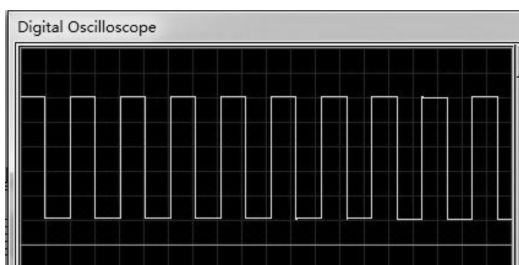


图 10-9 方波



## 10.6 源程序清单

```

KEY0      EQU      P1.0
KEY1      EQU      P1.1
KEY2      EQU      P1.2
SOUT      EQU      P2
FOUT      EQU      P1.3
FLAG      EQU      00H
SINP      EQU      30H
TH0D      EQU      32H
TL0D      EQU      33H

          ORG      0000H
          AJMP     MAIN
          ORG      000BH
          LJMP     INTT0_S
          ORG      0030H
MAIN:     MOV      SP, #60H
          MOV      SINP, #00H
          MOV      TMOD, #11H
          MOV      TH0D, #0FFH
          MOV      TL0D, #0AFH
          MOV      TH0, TH0D
          MOV      TL0, TL0D
          CLR      FLAG

CONZS:
          CPL      FLAG
          JB       FLAG, NOSIN
          MOV      DPTR, #TAB1

LOOP1:
          SETB     ET0
          SETB     EA
          SETB     TR0

LOOP:
          JNB      KEY0, INCKEY
          JNB      KEY1, DECKEY
          JNB      KEY2, CONSIN
          LJMP     LOOP

NOSIN:
          MOV      DPTR, #TAB2
          LJMP     LOOP1

; 按键 KEY0 功能, 输出频率增大
INCKEY:
          LCALL    DEL10MS
          JB       KEY0, LOOP1
          MOV      A, TL0D

```

```

        CJNE    A, #0AFH, INC1
        LJMP    LOOP1
INC1:    INC     TL0D
        LJMP    LOOP1

```

; 按键 KEY1 功能, 输出频率减小

DECKEY:

```

        LCALL   DEL10MS
        JB      KEY1, LOOP1
        MOV     A, TL0D
        CJNE    A, #00H, DEC1
        LJMP    LOOP1
DEC1:    DEC     TL0D
        LJMP    LOOP1

```

; 按键 KEY2 功能, 输出波形转换

```

CONSIN:  LCALL   DEL10MS
        JB      KEY2, LOOP
WAIT:     JNB     KEY2, WAIT
        LCALL   DEL10MS
        JNB     KEY2, WAIT
        LJMP    CONZS

```

; 定时器 T0 中断程序

INTT0\_S:

```

        PUSH    ACC
        CPL     FOUT
        MOV     TH0, TH0D
        MOV     TL0, TL0D
        MOV     A, SINP
        MOVC    A, @ A + DPTR
        MOV     SOUT, A
        INC     SINP
        POP     ACC
        RETI

```

; 10ms 延时程序

DEL10MS:

```

        MOV     R7, #0AH
DEL1:    MOV     R6, #0FFH
        DJNZ    R6, $
        DJNZ    R7, DEL1
        RET

```

; 正弦波函数表

```

TAB1:    DB 80H, 83H, 85H, 88H, 8AH, 8DH, 8FH, 92H
        DB 94H, 97H, 99H, 9BH, 93H, 0A0H, 0A3H, 0A5H
        DB 0A7H, 0AAH, 0ACH, 0AEH, 0B1H, 0B3H, 0B5H, 0B7H
        DB 0B9H, 0BBH, 0BDH, 0BFH, 0C1H, 0C3H, 0C5H, 0C7H
        DB 0C9H, 0CBH, 0CCH, 0CEH, 0D0H, 0D1H, 0D3H, 0D4H
        DB 0D6H, 0D8H, 0DAH, 0DBH, 0DCH, 0DDH, 0DEH
        DB 0DFH, 0E0H, 0E1H, 0E2H, 0E3H, 0E3H, 0E4H, 0E4H
        DB 0E5H, 0E5H, 0E6H, 0E6H, 0E7H, 0E7H, 0E7H, 0E7H
        DB 0E7H, 0E7H, 0E7H, 0E7H, 0E6H, 0E6H, 0E5H, 0E5H
        DB 0E4H, 0E4H, 0E3H, 0E3H, 0E2H, 0E1H, 0E0H, 0DFH

```

```

DB 0DEH, 0DDH, 0DCH, 0DBH, 0DAH, 0D8H, 0D7H, 0D6H
DB 0D4H, 0D3H, 0D1H, 0D0H, 0CEH, 0CCH, 0CBH, 0C9H
DB 0C7H, 0C5H, 0C3H, 0C1H, 0BFH, 0BDH, 0BBH, 0B9H
DB 0B7H, 0B5H, 0B3H, 0B1H, 0AEH, 0ACH, 0AAH, 0A7H
DB 0A5H, 0A3H, 0A0H, 9EH, 9BH, 99H, 97H, 94H
DB 92H, 8FH, 8DH, 8AH, 88H, 85H, 83H, 80H
DB 7DH, 7BH, 78H, 76H, 73H, 71H, 6EH, 6CH
DB 69H, 67H, 65H, 62H, 60H, 5DH, 5BH, 59H
DB 56H, 54H, 52H, 4FH, 4DH, 4BH, 49H, 47H
DB 45H, 43H, 41H, 3FH, 3DH, 3BH, 39H, 37H
DB 35H, 34H, 32H, 30H, 2FH, 2DH, 2CH, 2AH
DB 29H, 28H, 26H, 25H, 24H, 23H, 22H, 21H
DB 20H, 1FH, 1EH, 1DH, 1DH, 1CH, 1CH, 1BH
DB 1BH, 1AH, 1AH, 1AH, 19H, 19H, 19H, 19H
DB 19H, 19H, 19H, 19H, 1AH, 1AH, 1AH, 1BH
DB 1BH, 1CH, 1CH, 1DH, 1DH, 1EH, 1FH, 20H
DB 21H, 22H, 23H, 24H, 25H, 26H, 28H, 29H
DB 2AH, 2CH, 2DH, 2FH, 30H, 32H, 34H, 35H
DB 37H, 39H, 3BH, 3DH, 3FH, 41H, 43H, 45H
DB 47H, 49H, 4BH, 4DH, 4FH, 52H, 54H, 56H
DB 59H, 5BH, 5DH, 60H, 62H, 65H, 67H, 69H
DB 6CH, 6EH, 71H, 73H, 76H, 78H, 7BH, 7DH

```

；三角波函数表

TAB2:

```

DB 80H, 81H, 82H, 83H, 84H, 85H, 86H, 87H
DB 88H, 89H, 8AH, 8BH, 8CH, 8DH, 8EH, 8FH
DB 90H, 91H, 92H, 93H, 94H, 95H, 96H, 97H
DB 98H, 99H, 9AH, 9BH, 9CH, 9DH, 9EH, 9FH
DB 0A0H, 0A1H, 0A2H, 0A3H, 0A4H, 0A5H, 0A6H, 0A7H
DB 0A8H, 0A9H, 0AAH, 0ABH, 0ACH, 0ADH, 0AEH, 0AFH
DB 0B0H, 0B1H, 0B2H, 0B3H, 0B4H, 0B5H, 0B6H, 0B7H
DB 0B8H, 0B9H, 0BAH, 0BBH, 0BCH, 0BDH, 0BEH, 0BFH
DB 0BFH, 0BEH, 0BDH, 0BCH, 0BBH, 0BAH, 0B9H, 0B8H
DB 0B7H, 0B6H, 0B5H, 0B4H, 0B3H, 0B2H, 0B1H, 0B0H
DB 0AFH, 0AEH, 0ADH, 0ACH, 0ABH, 0AAH, 0A9H, 0A8H
DB 0A7H, 0A6H, 0A5H, 0A4H, 0A3H, 0A2H, 0A1H, 0A0H
DB 9FH, 9EH, 9DH, 9CH, 9BH, 9AH, 99H, 98H
DB 97H, 96H, 95H, 94H, 93H, 92H, 91H, 90H
DB 8FH, 8EH, 8DH, 8CH, 8BH, 8AH, 89H, 88H
DB 87H, 86H, 85H, 84H, 83H, 82H, 81H, 80H
DB 7FH, 7EH, 7DH, 7CH, 7BH, 7AH, 79H, 78H
DB 77H, 76H, 75H, 74H, 73H, 72H, 71H, 70H
DB 6FH, 6EH, 6DH, 6CH, 6BH, 6AH, 69H, 68H
DB 67H, 66H, 65H, 64H, 63H, 62H, 61H, 60H
DB 5FH, 5EH, 5DH, 5CH, 5BH, 5AH, 59H, 58H
DB 57H, 56H, 55H, 54H, 53H, 52H, 51H, 50H
DB 4FH, 4EH, 4DH, 4CH, 4BH, 4AH, 49H, 48H
DB 47H, 46H, 45H, 44H, 43H, 42H, 41H, 40H
DB 40H, 41H, 42H, 43H, 44H, 45H, 46H, 47H
DB 48H, 49H, 4AH, 4BH, 4CH, 4DH, 4EH, 4FH
DB 50H, 51H, 52H, 53H, 54H, 55H, 56H, 57H

```



```
DB  58H, 59H, 5AH, 5BH, 5CH, 5DH, 5EH, 5FH
DB  60H, 61H, 62H, 63H, 64H, 65H, 66H, 67H
DB  68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH
DB  70H, 71H, 72H, 73H, 74H, 75H, 76H, 77H
DB  78H, 79H, 7AH, 7BH, 7CH, 7DH, 7EH, 7FH
```

END



## 10.7 思考与练习

---

1. 利用模拟电路设计低频信号发生器。
2. 利用数字电路设计低频信号发生器。
3. 利用模拟电路、数字电路和 DAC0832 设计的低频信号发生器各有什么优缺点？
4. 基于图 10-3，考虑如何改变各种波形的频率。

# 第 11 章 基于 PCF8593 的实时时钟的设计与仿真

对于一些测控系统或手持式设备，经常需要显示和设定时间。目前，市场上有多种实时时钟提供了这类功能。这种可编程的实时时钟芯片内置了可编程的日历时钟及一定的 RAM 存储器，用于设定和保存时间。另外，实时时钟芯片一般内置闰年补偿系统，计时很准确。其采用备份电池供电，在系统断电时仍可工作。实时时钟芯片的这些优点，使得其广泛应用于需要提供时间数据及时间显示的场合。



## 11.1 系统功能

### 1. 概述

PCF8593 是低功耗的 CMOS 实时时钟/日历芯片，它提供一个可编程时钟输出，一个中断输出和掉电检测器，所有的地址和数据通过 I<sup>2</sup>C 总线接口串行传递。最大总线速度为 400Kb/s，每次读/写数据后，内嵌的字地址寄存器会自动产生增量。PCF8593 有 16 个 8 位寄存器：一个可自动增量的地址寄存器。一个内置 32.768kHz 的振荡器（带有一个内部集成的电容），一个分频器（用于给实时时钟 RTC 提供源时钟），一个可编程时钟输出，一个定时器，一个报警器，一个掉电检测器和一个 400kHz I<sup>2</sup>C 总线接口。

所有 16 个寄存器设计成可寻址的 8 位并行寄存器，但不是所有位都有用。前两个寄存器（内存地址 00H、01H）用于控制寄存器和状态寄存器，内存地址 02H ~ 08H 用于时钟计数器（秒~年计数器），地址 09H ~ 0CH 用于报警寄存器（定义报警条件），地址 0DH 控制 CLKOUT 引脚的输出频率，地址 0EH 和 0FH 分别用于定时器控制寄存器和定时器寄存器。秒、分钟、小时、日、月、年、分钟报警、小时报警、日报警寄存器，编码格式为 BCD，星期和星期报警寄存器不以 BCD 格式编码。

当一个 RTC 寄存器被读时，所有计数器的内容被锁存，因此，在传送条件下，可以禁止对时钟/日历芯片的错误读。

PCF8593 的引脚分布及描述如图 11-1 和表 11-1 所示。

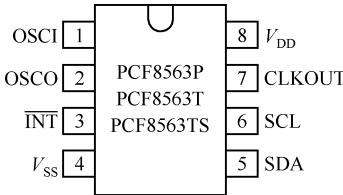


图 11-1 PCF8593 引脚分布

表 11-1 PCF8593 引脚描述

符 号	引 脚 号	描 述
OSCI	1	振荡器输入
OSCO	2	振荡器输出
INT	3	中断输出（开漏），低电平有效
V <sub>ss</sub>	4	地

续表

符 号	引 脚 号	描 述
SDA	5	串行数据 I/O
SCL	6	串行时钟输入
CLKOUT	7	时钟输出 (开漏)
$V_{DD}$	8	正电源

## 2. 特性

- 低工作电流：典型值为  $0.25\mu\text{A}$  ( $V_{DD} = 3.0\text{V}$ ,  $T_{amb} = 25^\circ\text{C}$  时)。
- 世纪标志。
- 大工作电压范围： $1.0 \sim 5.5$ 。
- 低休眠电流：典型值为  $0.25\mu\text{A}$  ( $V_{DD} = 3.0\text{V}$ ,  $T_{amb} = 25^\circ\text{C}$ )。
- $400\text{kHz}$  的  $I^2C$  总线接口 ( $V_{DD} = 1.8 \sim 5.5\text{V}$  时)。
- 可编程时钟输出频率为  $32.768\text{kHz}$ 、 $1024\text{Hz}$ 、 $32\text{Hz}$ 、 $1\text{Hz}$ 。
- 报警和定时器。
- 掉电检测器。
- 内部集成的振荡器电容。
- 片内电源复位功能。
- $I^2C$  总线从地址：读， $0A3H$ ；写， $0A2H$ 。
- 开漏中断引脚。

## 3. 应用

- 移动电话。
- 便携仪器。
- 传真机。
- 电池电源产品。



## 11.2 设计方案

本例中主要使用 AT89C51、74HC164、IEC、PCF8593、7SEG-COM-AN-GRN 等器件组成硬件电路，其中 PCF8593 用于读取系统时间，AT89C51、74HC164、IEC、7SEG-COM-AN-GRN 等器件用于显示 PCF8593 读取的系统时间，从而体现 PCF8593 的功能。



## 11.3 系统硬件仿真电路

硬件电路连接如图 11-2、图 11-3 和图 11-4 所示。

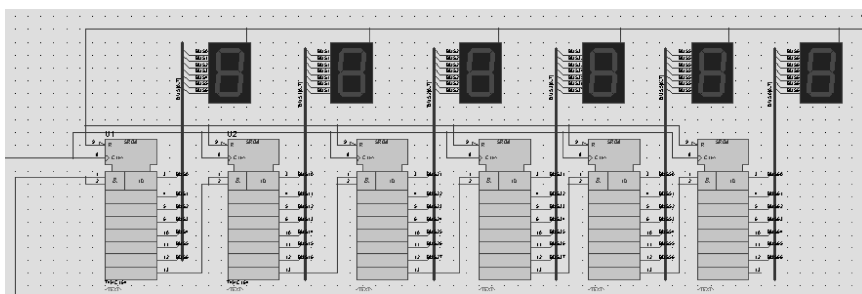


图 11-2 显示系统时间电路图

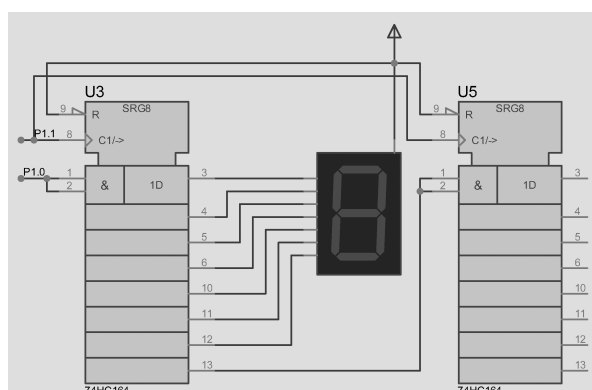


图 11-3 74HC164 与 7 段数码管连接图

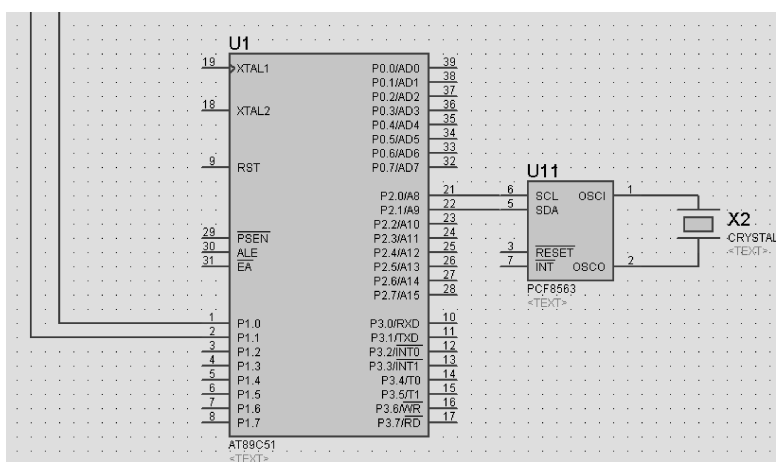


图 11-4 读取系统时间电路图



## 11.4 程序设计

该程序的主流程图如图 11-5 所示。

其中，AT89C51 读取 PCF8593 的时钟信号子程序流程图如图 11-6 所示。

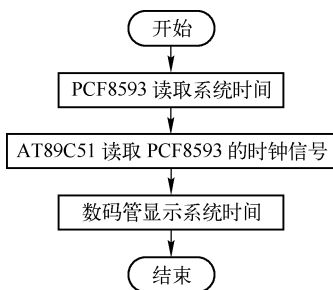


图 11-5 程序主流程图

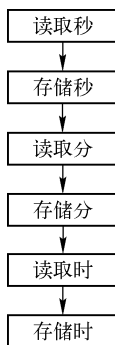


图 11-6 读取时钟信号子程序流程图



## 11.5 软件调试与运行结果

按照前面介绍的进行调试，运行结果如图 11-7 所示。

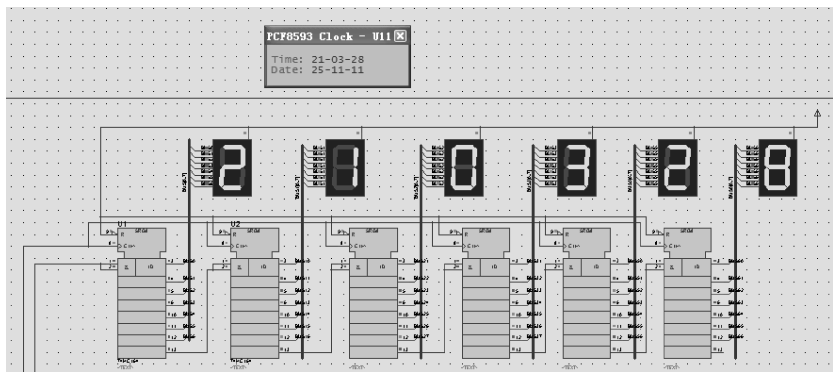


图 11-7 运行结果



## 11.6 PCF8593 精度的调整

仿真过程中，PCF8593 直接读取计算机系统的时间。但在实际应用中，是需要其自身提供时间数据的，经常会出现精度不够的问题。如图 11-3 所示的 PCF8593 应用电路，为图中晶振的两端添加石英晶片并对其频率进行调整，可以提高精度，具体有以下 3 种可行的方法。

(1) 定值 OSCI 电容。计算出所需的电容平均值，用此值的定值电容，通电后在 CLK-OUT 引脚上测出的频率应为 32.768kHz。

(2) OSCI 微调电容。可通过调整 OSCI 引脚的微调电容式振荡器的频率来获得更高的精度，此时可测出通电时引脚上的信号频率为 32.768kHz。

(3) OSCI 输出。直接测量 OSCI 引脚的输出。



## 11.7 源程序清单

该程序主要由 I<sup>2</sup>C 协议和主程序两部分组成。

### 1. I<sup>2</sup>C 协议代码

```

;      MCS-51 单片机模拟 I2C 软件包
;功能说明: 本模拟 I2C 软件包包含 I2C 操作的底层子程序, 使用前要定义好 SCL 和 SDA。在标准 8051 模式(12Clock)下, 对主频要求是不高于 12MHz(即 1 个机器周期 1μs); 若  $F_{osc} > 12\text{MHz}$ , 则要增加相应的 NOP 指令数(总线时序符合; I2C 标准模式, 100Kb/s)
;启动 I2C 总线子程序
START:  SETB  SDA
        NOP
        SETB  SCL           ;起始条件建立时间大于 4.7μs
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        CLR   SDA
        NOP           ;起始条件锁定时大于 4μs
        NOP
        NOP
        NOP
        NOP
        CLR   SCL           ;钳住总线, 准备发数据
        NOP
        RET

;结束总线子程序
STOP:   CLR   SDA
        NOP
        SETB  SCL           ;发送结束条件的时钟信号
        NOP           ;结束总线时间大于 4μs
        NOP
        NOP
        NOP
        NOP
        SETB  SDA           ;结束总线
        NOP           ;保证一个终止信号和起始信号的空闲时间大于 4.7μs
        NOP
        NOP
        NOP
        RET

;发送应答信号子程序
MACK:   CLR   SDA           ;将 SDA 置 0

```

```

NOP
NOP
SETB  SCL
NOP           ;保持数据时间, 即 SCL 为高时间大于 4.7μs
NOP
NOP
NOP
NOP
CLR  SCL
NOP
NOP
RET
;发送非应答信号
MNACK: SETB  SDA           ;将 SDA 置 1
NOP
NOP
SETB  SCL
NOP
NOP           ;保持数据时间, 即 SCL 为高时间大于 4.7μs
NOP
NOP
NOP
CLR  SCL
NOP
NOP
RET
;检查应答位子程序
;返回值, ACK = 1 时表示有应答
CACK: SETB  SDA
NOP
NOP
SETB  SCL
CLR  ACK
NOP
NOP
MOV  C, SDA
JC   CEND
SETB ACK           ;判断应答位
CEND: NOP
CLR  SCL
NOP
RET
;发送字节子程序
;字节数据放入 ACC
;每发送一字节要调用一次 CACK 子程序, 取应答位
WRBYTE: MOV  R0, #08H
WLP:    RLC  A           ;取数据位
JC      WR1
SJMP   WR0           ;判断数据位
WLP1:   DJNZ R0, WLP

```





```

NOP
NOP
NOP                ; +1
NOP
NOP                ; +1
NOP                ; +1
DJNZ R0,RLP        ;未够 8 位, 再来一次
RET

;以下是用户接口子程序
;无子地址器件写字节数据
;入口参数: 数据为 ACC
;占用: A、R0、CY
IWRBYTE:
IWBLOOP: LCALL START                ;启动总线
          LCALL WRBYTE
          LCALL CACK
          LCALL STOP
          RET
RETWRB:   LCALL STOP
          RET

;无子地址器件读字节数据
;出口参数: 数据为 ACC
;占用: A、R0、R2、CY
IRDBYTE:  LCALL START
          LCALL RDBYTE                ;进行读字节操作
          LCALL MNACK                ;发送非应答信号
RETRDB:   LCALL STOP                ;结束总线
          RET

;向器件指定子地址写 N 字节数据
;入口参数: 器件子地址 SUBA、发送数据缓冲区 MTD、发送字节数 NUMBYTE
;占用: A、R0、R1、R3、CY
IWRNBYTE: MOV A, NUMBYTE
          MOV R3, A
          LCALL START                ;启动总线
          MOV A, SUBA                ;指定子地址
          LCALL WRBYTE
          LCALL CACK
          MOV R1, #MTD
WRDA:     MOV A, @ R1
          LCALL WRBYTE                ;开始写入数据
          LCALL CACK
          JNB ACK, IWRNBYTE
          INC R1
          DJNZ R3, WRDA                ;判断是否写完
RETWRN:   LCALL STOP
          RET

;向器件指定子地址读取 N 字节数据
;入口参数: 器件子地址 SUBA、接收字节数 NUMBYTE
;出口参数: 接收数据缓冲区 MTD
;占用: A、R0、R1、R2、R3、CY

```

```

IRDNBYTE: MOV    R3,NUMBYTE
           LCALL  START
           MOV    A,SUBA                ;指定子地址
           LCALL  WRBYTE
           LCALL  CACK
           LCALL  START                ;重新启动总线
           MOV    A,SLA
           INC    A;                    准备进行读操作
           LCALL  WRBYTE
           LCALL  CACK
           JNB    ACK,IRDNBYTE
           MOV    R1,#MRD
RDN1:     LCALL  RDBYTE                ;读操作开始
           MOV    @ R1,A
           NOP                                ; +1
           NOP                                ; +1
           DJNZ   R3,SACK
           LCALL  MNACK                ;最后一字节发非应答位
RETRDN:   LCALL  STOP                ;并结束总线
           RET
SACK:     LCALL  MACK
           INC    R1
           SJMP   RDN1
;无子地址器件写 N 字节数据
;入口参数: 发送数据缓冲区 MTD、发送字节数 NUMBYTE
;占用:A、R0、R1、R3、CY
;不发送子地址 SUBA, 直接发送多个数据
IWRNBYTEEXT:
           MOV    A,NUMBYTE
           MOV    R3,A
           LCALL  START                ;启动总线
           MOV    R1,#MTD
WRDAE:    MOV    A,@ R1
           LCALL  WRBYTE                ;开始写入数据
           LCALL  CACK
           JNB    ACK,RETRWRNE
           INC    R1
           DJNZ   R3,WRDAE            ;判断是否写完
RETRWRNE: LCALL  STOP
           RET
;无子地址器件读取 N 字节数据
;入口参数: 器件子地址 SUBA、接收字节数 NUMBYTE
;出口参数: 接收数据缓冲区 MTD
;占用:A、R0、R1、R2、R3、CY
;说明:此函数不发送子地址,也不重新启动总线,而是直接读取多个数据
IRDNBYTEEXT:
           MOV    R3,NUMBYTE
           LCALL  START
           MOV    R1,#MRD
RDNE1:    LCALL  RDBYTE                ;读操作开始

```

```

MOV    @ R1,A
DJNZ   R3,SACKE
LCALL  MNACK                      ;最后一字节发非应答位
RETRDNE: LCALL  STOP              ;并结束总线
RET
SACKE:  LCALL  MACK
INC     R1
SJMP    RDNE1

```

;请注意

;占用内部资源:R0, R1, R2, R3, ACC, Cy

;在你的程序里要做以下定义:

;(1)定义变量: SUBA 器件子地址、NUMBYTE 读/写的字节数、位变量 ACK

;(2)定义常量: SDA SCL 总线位、MTD 发送数据缓冲区首址、MRD 接收数据缓冲区首址

;(ACK 为调试/测试位, ACK 为 0 时表示无器件应答或总线出错)

;可以对“用户接口子程序”进行裁减, 即把不使用的接口子程序删掉, 以节省程序空间

## 2. 主程序

```

SDA    BITP2.1;      I2C 总线定义
SCL     BITP2.0
MTD     EQU40H        ;发送数据缓冲器
MRD     EQU49H        ;接收数据缓冲区
;定义器件地址
ACK     BIT10H        ;应答标志位
SUBA    DATA 51H     ;器件的子地址
NUMBYTE DATA 52H     ;读/写的字节数变量
ORG     0000H
LJMP    MAIN
ORG     0100H
;----- 主程序如下
MAIN:
MOV     SP,#70H
LCALL   DELAY
MOV     MTD,#00H
MOV     MTD+1,#12H
MOV     MTD+2,#00H
MOV     MTD+3,#30H
MOV     MTD+4,#12H
MOV     MTD+5,#
MOV     SUBA,#00H      ;指定子地址
MOV     NUMBYTE,#06H   ;发送 6 字节数据
LCALL   IWRNBYTE       ;调用写 6 字节数据程序
LCALL   DELAY
MAINLOOP:
MOV     SUBA,#06H      ;指定子地址
MOV     NUMBYTE,#06H   ;发送 6 字节数据
LCALL   IRDNBYTE       ;调用读 6 字节数据程序
LCALL   DISPLAY
LCALL   DELAY

```

```

        SJMP    MAINLOOP

DELAY:
        MOV     R7,#00H
MIN:    DJNZ    R7,YS500
        RET
YS500:  LCALL   YS500US
        LJMP    MIN
YS500US: MOV     R6,#00H
        DJNZ    R6,$
        RET
DELAY1: MOV     R7,#20H
        DJNZ    R7,$
        RET
DISPLAY:
        MOV     R0,#49H
DISP_LOOP:
        MOV     A,@ R0
        LCALL   TRANS
        INC     R0
        CJNE    R0,#5DH,DISP_LOOP
        RET
TRANS:  MOV     DPTR,#TAB
        MOVC    A,@ A + DPTR
        MOV     R4,#08H
TRANS_LOOP:
        RRC     A
        MOV     P1.0,C
        SETB    P1.1
        CLR     P1.1
        DJNZ    R4,TRANS_LOOP
        RET
TAB:    DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H
        DB 7FH,6FH,77H,7CH,39H,5EH,79H,71H
        DB 00H

```



## 11.8 思考与练习

1. 在手持式设备应用中，如何保持实时时钟的低功耗？
2. 让 PCF8593 在每小时的 45 分钟时产生报警并在  $\overline{\text{INT}}$  端产生一个中断给单片机。
3. 让 PCF8593 每秒钟产生一次报警并在  $\overline{\text{INT}}$  端产生一个脉冲给单片机。在中断服务程序中可以读取时钟以供显示。
4. 在 PCF8593 的 CLKOUT 引脚输出一个 32.768kHz 的方波。

# 第 12 章 主从式多单片机通信系统的设计与仿真

随着计算机技术的发展及工业自动化水平的提高，在许多场合采用单机控制已不能满足现场要求，因而必须采用多机控制形式，而多机控制主要通过多个单片机之间的串行通信实现。串行通信作为单片机之间常用的通信方法之一，由于其通信编程灵活、硬件简单并遵行统一的标准，因此在工业控制领域得到了广泛的应用。主从式多机通信就是其中的一种多机通信方式。这种多机系统结构简单，应用广泛，但只能实现由主机呼叫分级，然后实现主机与分机之间的全双工串行通信。



## 12.1 设计方案

3 个 AT89C51 单片机之间进行“1 主 2 从”的多级通信，主机可以将其数码管显示的内容发送给每个从机，也可以采集每个从机数码管显示的数值并求和后显示出来。每个单片机的数码管显示值可以通过外界的按键进行设置。



## 12.2 系统硬件仿真电路

### 1. 元器件清单列表

打开 PROTEUS ISIS 编辑环境，按表 12-1 所列清单添加元器件。

表 12-1 元器件清单

元器件名称	所 属 类	所 属 子 类
AT89C51	Microprocessor Ics	MCS8051
7SEG-BCD	Optoelectronics	Display
BUTTON	Switches & Relays	Active
MINRES10K	Resistors	Resistors

### 2. 电路原理图

元器件全部添加后，在 PROTEUS ISIS 的编辑区域中按图 12-1 和图 12-2 所示的主、从机电路原理图连接硬件电路。

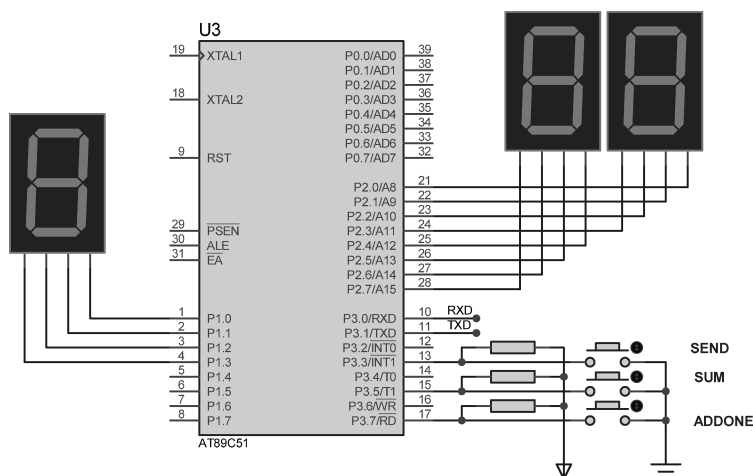


图 12-1 主机电路原理图

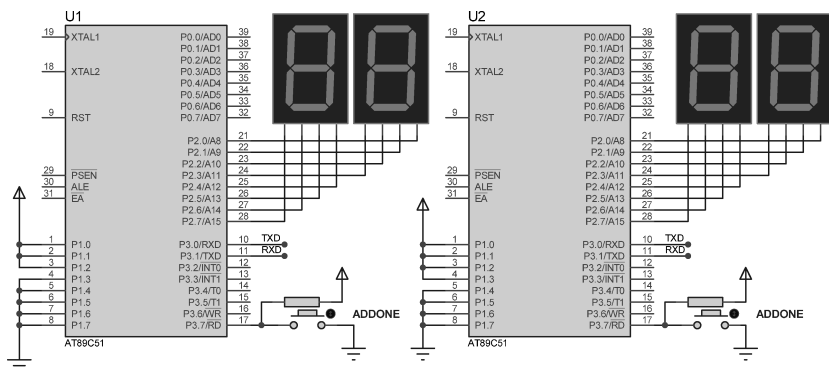


图 12-2 从机电路原理图



## 12.3 程序设计

### 1. 主机部分

主机部分程序主要包括按键检测、状态显示、数值显示和数据发送与接收等部分。按键检测分为三部分：主机向从机发送数据按键、从机数据求和按键及主机自身数值 +1 按键。按图 12-1 所示分别应用 I/O 口 P3.3、P3.5 和 P3.7 完成。状态显示状态值，状态值内容也分为三部分：当数值显示内容为主机自身数值时，状态值为“7”；当数值显示内容为两个从机数据之和时，状态值为“5”；当主机自身的数值发送给从机时，状态值为“3”。数值显示分为自身数据显示、从机求和数据显示及向从机发送数据显示三部分。数据发送与接收是指通过串行口向从机发送和从从机读取数据。


### 2. 从机部分

从机部分程序包括机号检测、数值显示、数据发送与接收和按键检测等部分。机号检测是利用 P1 口的状态代表机器号。数值显示包括两部分：一是自身的数值，二是主机发送的

数值。数据发送与接收是收到主机的发送或接收命令时向主机发送数据、接收主机发送的数据。按键检测是显示自身数值时，每按一次自身“+1”按键数据+1的功能。



## 12.4 软件调试与运行结果

参照 9.1 节建立程序文件，分别加载目标代码文件到 3 个单片机上，在 PROTEUS ISIS 界面中，单击  按钮启动仿真。

### 1) 主机操作

- (1) 每按下“+1”键，数码管显示值加 1，对应 P1 口的数码管显示“7”。
- (2) 每按下“SUM”键，主机对应 P2 口的数码管显示值变为 1 号从机的显示值与 2 号从机显示值之和，主机对应 P1 口的数码管显示“5”。
- (3) 每按下“SEND”键，各从机的数码管显示值均变为主机 P2 口对应数码管显示的数值，主机对应 P1 口的数码管显示“3”。

### 2) 从机操作

- (1) 每按下“+1”键，数码管显示值加 1。
- (2) 运行中的数码管显示值随主机的操作而发生变化。
- (3) 仿真运行片段如图 12-3 和图 12-4 所示。仿真过程中可单击按钮暂停仿真，从“Debug”菜单中调出各个单片机的“8051 CPU Registers”窗口来观察各单片机运行中相关寄存器的工作状态，如图 12-5 所示为主单片机相关寄存器的工作状态。

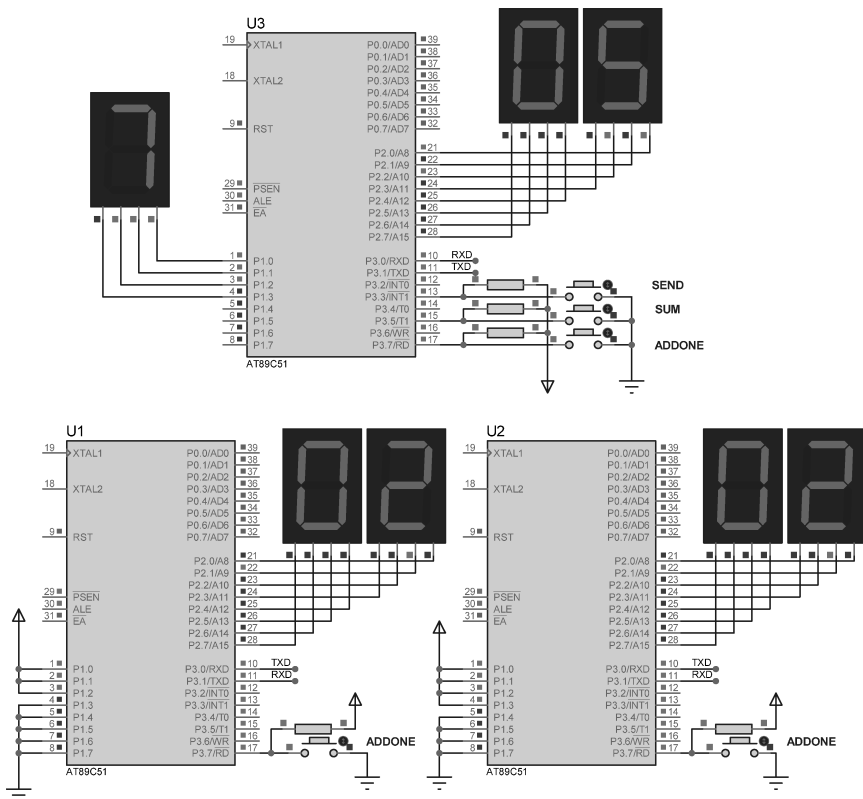


图 12-3 仿真运行片段 1

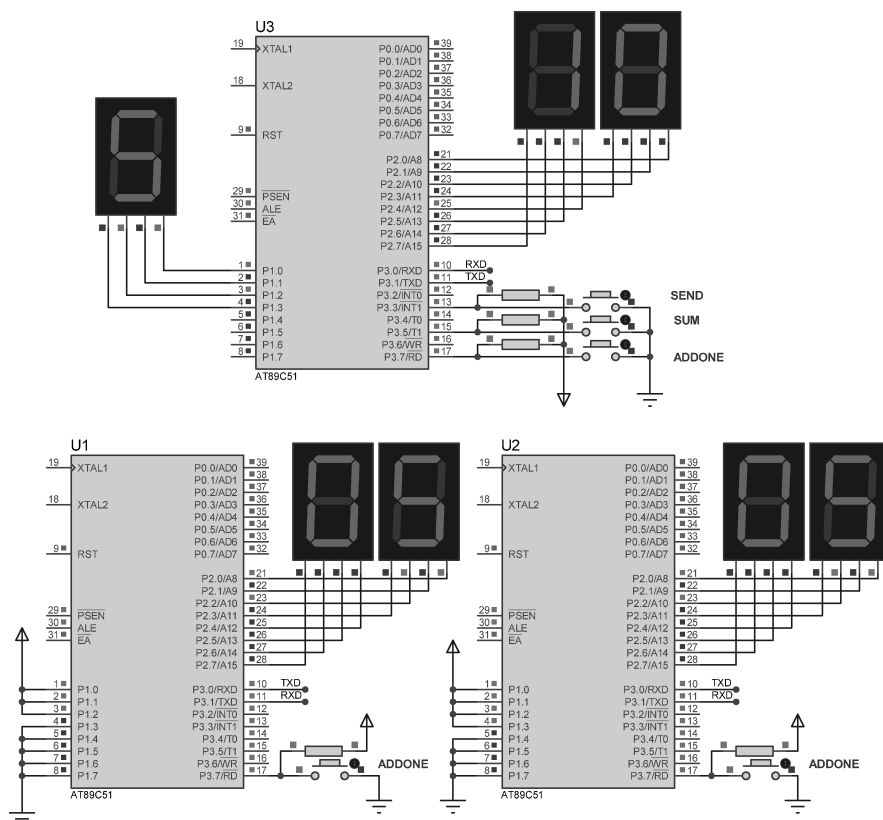


图 12-4 仿真运行片段 2

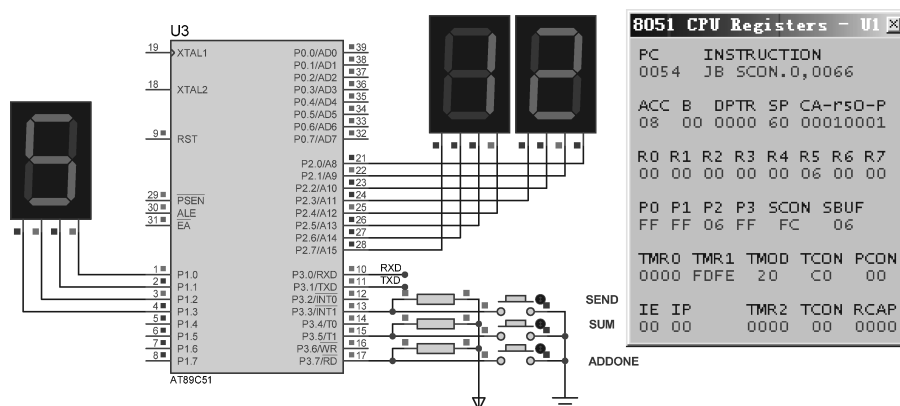


图 12-5 主单片机相关寄存器的工作状态



## 12.5 源程序清单

### 1. 主机部分

```
ORG    0000H
AJMP   MAIN
```



```

                ORG      0030H
MAIN:           MOV      SP,#60H
                MOV      TMOD,#20H
                MOV      TH1,#0FDH
                MOV      TL1,#0FDH
                MOV      SCON,#0D8H
                MOV      PCON,#00H
                SETB     TR1
                MOV      R5,#00H
                MOV      P1,#00H
LOOP:           MOV      P2,R5
                MOV      A,#0FFH
                MOV      P3,A
                JNB      P3.3,SEND
                JNB      P3.5,SUM
                JNB      P3.7,ADDONE
                LJMP     LOOP
SEND:           JNB      P3.3,$
                MOV      P1,#03H
                MOV      R3,#03H
                MOV      R2,#07H
                LCALL    MSIO1
                NOP
                NOP
                MOV      R2,#0FH
                LCALL    MSIO1
                LJMP     LOOP
SUN:            JNB      P3.5,$
                MOV      P1,#05H
                MOV      R3,#01H
                MOV      R2,#07H
                LCALL    MSIO1
                NOP
                NOP
                MOV      A,R5
                MOV      R4,A
                MOV      R2,#0FH
                LCALL    MSIO1
                MOV      A,R4
                ADD      A,R5
                DA        A
                MOV      R5,A
                LJMP     LOOP
ADDONE:         JNB      P3.7,$
                MOV      P1,#07H
                INC      R5
                CLR      A
                ADD      A,R5
                DA        A
                MOV      R5,A

```

```

        LJMP    LOOP
MSIO1:  SETB    TB8
        MOV     A,R2
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        CLR     TB8
        MOV     A,R3
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        CJNE    A,#00H,RECE
        CLR     TB8
        MOV     A,R5
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        RET
RECE:   JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     R5,A
        RET
END

```

## 2. 从机部分

```

SLAVE      EQU 2FH
ORG         0000H
AJMP        MAIN
ORG         0030H
MAIN:       MOV     SP,#60H
            MOV     TMOD,#20H
            MOV     TH1,#0FDH
            MOV     TL1,#0FDH
            MOV     SCON,#0D8H
            MOV     PCON,#00H
            SETB    TR1
            SETB    SM2
            MOV     A,#0FFH
            MOV     P1,A
            MOV     A,P1
            MOV     SLAVE,A
            MOV     R5,#00H
            SETB    P3.7
DISPLAY:    MOV     P2,R5
CHKRI:     JB      RI,SSIO
            JB      P3.7,CHKRI
            JNB     P3.7,$

```

```

        INC     R5
        MOV     A,#00H
        ADD     A,R5
        DA      A
        MOV     R5,A
        LJMP    DISPLAY
SSIO:   CLR     RI
        SETB    RS1
        CLR     RS0
        MOV     A,SBUF
        XRL     A,SLAVE
        JZ      SSIO1
RETURN: SETB    SM2
        LJMP    DISPLAY
SSIO1:  CLR     SM2
        JNB     RI,$
        CLR     RI
SSIO2:  MOV     A,SBUF
        CJNE    A,#00H,SEND
        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     R5,A
        LJMP    RETURN
SNED:   MOV     A,R5
        MOV     SUBF,A
        JNB     TI,$
        CLR     TI
        LJMP    TETURN
        END

```



## 12.6 思考与练习

1. 在多机通信应用中，主机、从机的串行口工作方式如何设置？
2. 把从机串行口查询工作方式更改为中断工作方式，如何进行编程？
3. 如果不采用主从式结构，还有其它方法吗？如何应用？
4. 结合高级语言编程设计上位机软件，实现计算机和多单片机的数据通信。

# 第 13 章 基于 PID 控制算法的温度控制系统的设计与仿真

电阻炉在化工、冶金等行业应用广泛，因此温度控制在工业生产和科学研究中具有重要意义。其控制系统属于一阶纯滞后环节，具有大惯性、纯滞后、非线性等特点，导致传统控制方式超调大、调节时间长、控制精度低。采用单片机进行炉温控制，具有电路设计简单、精度高、控制效果好等优点，对提高生产效率、促进科技进步等具有重要的现实意义。



## 13.1 设计方案

在温度测量控制系统中，实际温度值由 PT100 恒流工作调理电路进行测量。为了克服 PT100 线性度不好的缺点，在信号调理电路中加入负反馈非线性校正网络；调理电路的输出电压经 ADC0808 转换后送入单片机 AT89C52；对采样数据进行滤波及标定处理后，由 3 位 7 段数码管显示。输入的设定值由 4 位独立按键电路进行设定，可分别对设定值的十位和个位进行加 1、减 1 操作。设定值送入单片机后，由另外一组 3 位 7 段数码管显示。数码管的段码由 74HC05 驱动，位码由三极管 2N222A 驱动。为了使两组数码管实时显示，对两组数码管进行动态扫描。整体的电路原理框图如图 13-1 所示。

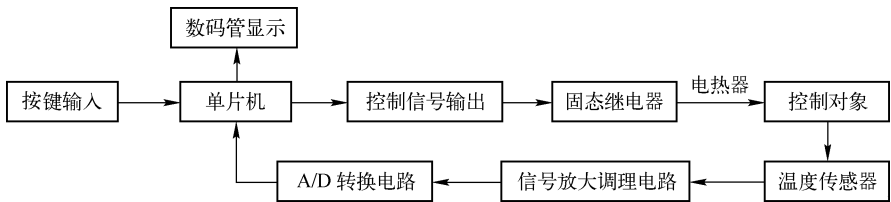


图 13-1 整体电路原理框图

系统采用 PID 闭环控制方案。如图 13-2 所示，将预置初值与温度传感器反馈信号比较得到偏差 ( $e$ )，对偏差 ( $e$ ) 进行 PID 运算处理得到控制量 ( $u$ )，通过此量来控制加热器的加热时间，从而控制加热功率。由于水本身具有很大的热惯性，所以必须对水温的变化趋势做出预测，并且根据需要及时反方向抑制，以防止出现较大的超调量和波动。在 PID 控制中，积分环节 (I) 具有很强的滞后效应，而微分环节 (D) 具有预见性，所以该方案最终采用 PD 算法，能够很好地控制超调，并且稳态误差也很小。

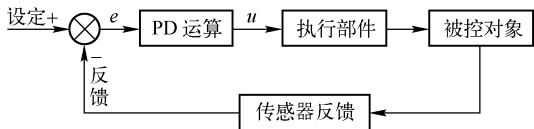


图 13-2 系统控制方案



## 13.2 系统硬件仿真电路

### 1. 温度测量调理电路 (见图 13-3)

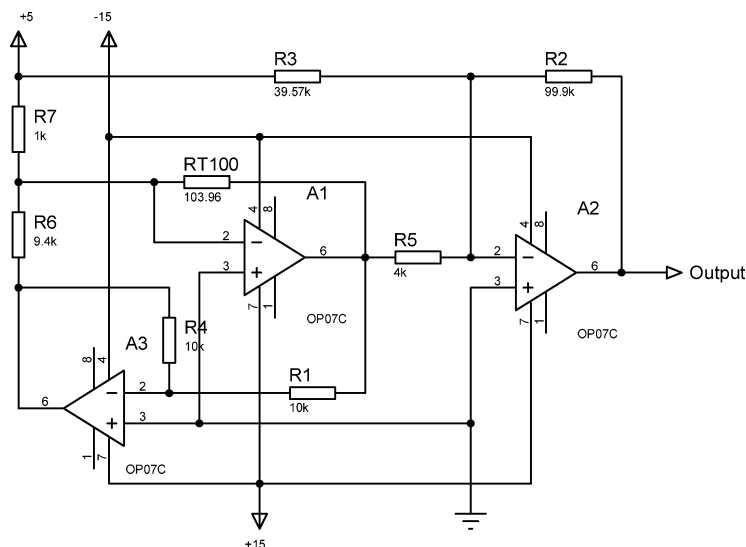


图 13-3 温度测量调理电路

本系统采用恒流工作调理电路，铂电阻选用标称值为  $100\Omega$  的 PT100 作为温度传感器。A1、A2、A3 采用低漂移运放 OP07C，由于有电流流经 PT100 传感器，所以当温度为  $0^\circ\text{C}$  时，在 PT100 传感器上有电压降，这个电压为 PT100 传感器的偏置电压，是运放 A1 输出电压的一部分，使恒流工作调理电路的输出实际不为零。所以需要对这个偏置电压调零，R3 为调零电阻，其作用为当温度为  $0^\circ\text{C}$  时，将恒流工作调理电路的输出调到零。又因为 PT100 的电阻特性为非线性，PT100 在  $0 \sim 100^\circ\text{C}$  变化范围内非线性误差为  $0.4\%$  ( $0.4^\circ\text{C}$ )，由于本系统无小数显示， $0.4^\circ\text{C}$  的误差本身不会对 A/D 量化和数码管显示造成影响，但由于软件编制中，对标度变化程序中的变换系数作了近似处理，使得非线性误差接近  $0.79\%$  ( $0.79^\circ\text{C}$ )，就有可能对 A/D 量化和数码管显示造成影响，所以加进了线性化电路，运放 A3 及电阻 R1、R4 和 R6 一同构成了负反馈非线性校正网络。R5 用于调整运放 A2 的增益。

电路的调整方法如下（用普通电阻代替 PT100 进行调整）。

- 接入相当于  $0^\circ\text{C}$  的  $100\Omega$  的电阻，用于 R3 调零。
- 接入相当于  $50^\circ\text{C}$  的  $119.70\Omega$  的电阻，用于 R5 调整增益。
- 接入相当于  $100^\circ\text{C}$  的  $139.10\Omega$  的电阻，用于 R1 或 R4 调整线性。

反复调整多次，在  $0 \sim 100^\circ\text{C}$  温度范围内适宜为止。

经上述信号电路调理，显示对照表如表 13-1 所列。

以温度值为横坐标，电压值为纵坐标，由表 13-1 分析非线性误差可知：在  $50^\circ\text{C}$  时，存在最大偏差为  $0.005^\circ\text{C}$ ，故非线性校正后非线性误差变为  $0.1\%$  ( $0.1^\circ\text{C}$ )，A/D 量化及数码管显示不会产生误操作。

表 13-1 显示对照表

理想温度 值/℃	0	10	20	30	40	50	60	70	80	90	100
对应电阻 值/Ω	100	103.96	107.91	111.85	115.78	119.70	123.49	127.49	131.37	135.24	139.10
实际输出 电压/V	0.002	0.502	1.004	1.502	2.004	2.505	3.001	3.502	4.001	4.498	4.996
显示温度 值/℃	0	10	20	30	40	50	60	70	80	90	100

## 2. A/D 转换电路

本系统采用 5V 的电压源, 用 PT100 电阻传感器组成的信号调理电路作为信号的输入装置, 当 PT100 传感器置于温度场时, 调理电路将根据 PT100 的阻值输出相应的电压值。将该输出电压送到 ADC0808 的模拟量输入通道 IN<sub>0</sub>, 经 ADC0808 进行模数转换, 将标准的模拟信号转换为等价的数字信号。本设计选用 IN<sub>0</sub> 作为模拟量输入通道, 则将 ADC0808 的 A、B、C 三条地址线均置为低电平。转换启动信号 START 接到 AT89C52 的 P2.0 口, 转换结束状态信号 EOC 接 P3.7 口, 输出允许信号 OE 接 P3.6 口, 地址锁存允许信号 ALE 接 P3.3 口, 由于 ADC0808 内部没有时钟电路, 所以用 AT89C52 的 ALE 经二分频接 ADC0808 的 CLK 端, VREF- 接地, VREF+ 接 +5V 电压。ADC0808 的转换结果输出到 AT89C52 的 P0.0 ~ P0.7 口, 作为 AT89C52 的输入信号。A/D 转换接口电路如图 13-4 所示。

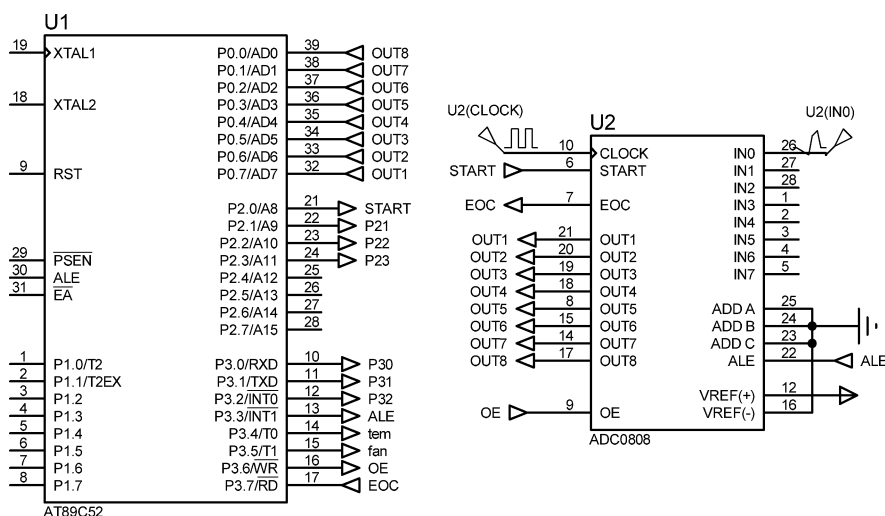


图 13-4 A/D 转换接口电路

### 3. 按键输入电路

本系统采用 4 个按键搭建键盘电路，如图 13-5 所示。第一个按键用来判断是转入控制处理子程序运行，还是转入键盘处理子程序运行。若未按下则转入控制处理子程序运行，按下则转入键盘处理子程序运行；若第一个按键按下，则第二个按键开始起作用，用第二个按

键来判断是十位进行加减操作，还是个位进行加减操作。若第二个按键未按下，转十位进行加减操作，否则转个位进行加减操作；第三个按键为减一操作，第四个按键为加一操作。为了方便按键操作，将个位和十位的设定值均设置为 5，如果加一操作结果等于 11，给加一单元重新赋值 5，如果减一操作结果等于 0FFH，给减一单元重新赋值 5。这样考虑最坏情况，即用键盘设置离初始设定值最远的值，第三个按键最多按 5 次，第四个按键最多按 5 次。从而大大减少了按键次数，且更方便地给出设定值。第一个键和第二个键的加入，也充分考虑了总程序的整体调度。

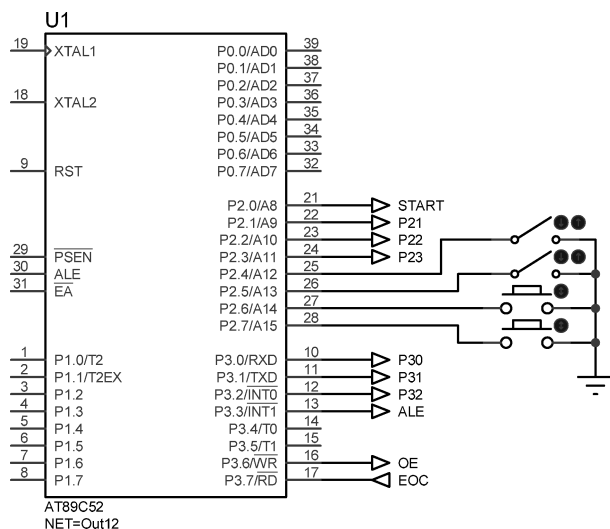


图 13-5 按键输入电路

#### 4. 数码管显示电路

显示电路采用两个 4 位 LED 显示数码管，共阴极接法。第一个数码管显示 A/D 转换数据采集的采样值，选用 3 位显示采样值，显示范围为 0 ~ 100；第二个数码管显示由键盘输入的设定值，用于显示对系统的温度设定，也选用 3 位显示设定值，显示范围为 0 ~ 100。由于 LED 显示电路较多选用动态扫描方式，为了实现 LED 显示器的动态扫描，除了要给显示器提供段的输入外，还要对显示器的位进行控制，即位控和位控。所以需要 P1 口输出 8 条段控线；位控线由挑选的 P2.1、P2.2、P2.3、P3.0、P3.1 和 P3.2 输出，其中 P2.1、P2.2 和 P2.3 用于驱动键盘输入的设定值的数码管位控线，其余用于驱动显示 A/D 转换数码管的数据码位控线，位控线的数目等于数码管显示的位数。

P1 接口最多可连接 8 个 LED 显示器。为提高显示亮度，通常加 74HC05 进行段控输出驱动，与 7 段数码管的段码驱动输入端相连，由于位控位的驱动电流较大，8 段全亮需 40 ~ 60mA，所以用三极管 9012 提高驱动能力，其集电极接到 7 段数码管的位码驱动输入端，三极管的发射极接地，将 AT89C52 的 P3.0、P3.1、P3.2 分别与一个 2kΩ 的电阻连接到三极管的基极，用于驱动采样值显示数码管，将 AT89C52 的 P2.0、P2.1 和 P2.2 口分别与一个 2kΩ 的电阻连接到三极管的基极，用于驱动设定值显示数码管。显示电路如图 13-6 所示。

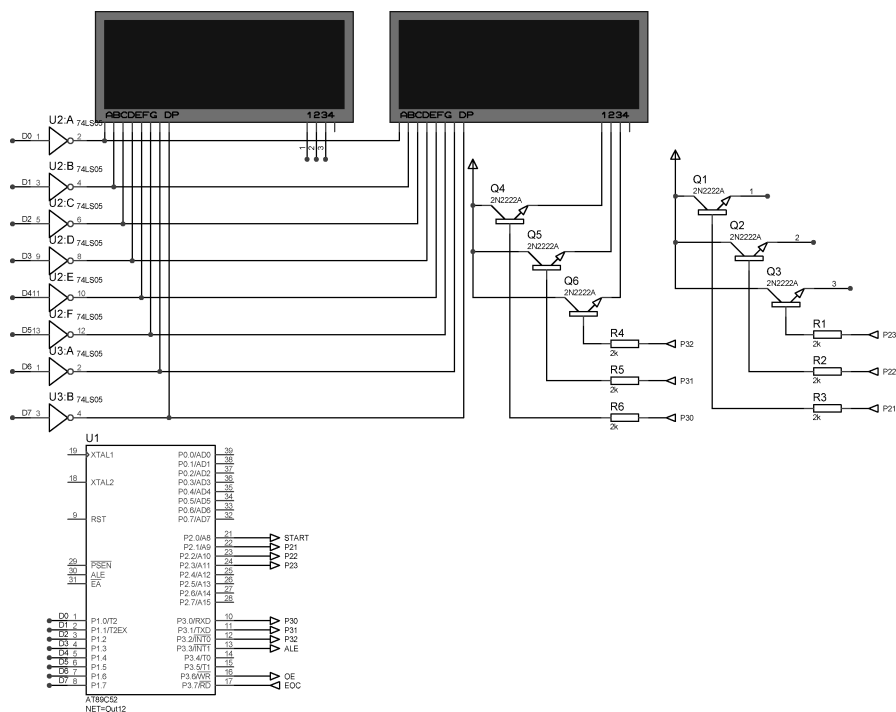


图 13-6 显示电路

### 5. 温度控制电路

系统的电阻丝和风扇均采用如图 13-7 所示的电路形式。此电路采用晶体管驱动固态继电器。当 P3.4 为低电平时，继电器 RL1 吸合；当 P3.4 或 P3.5 为高电平时，继电器 RL1 释放。采用这种控制逻辑可以使继电器在上电复位或单片机复位时不吸合。继电器由晶体管 2N2222A 驱动，它可以提供所需的驱动电流。

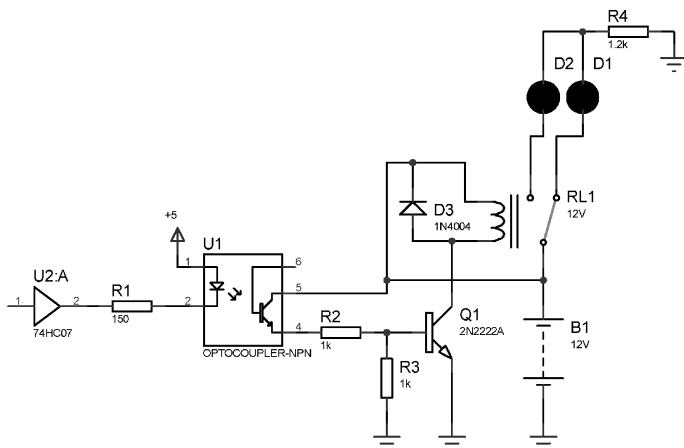


图 13-7 继电器控制电路

PT100 温度测控系统的完整电路如图 13-8 所示。



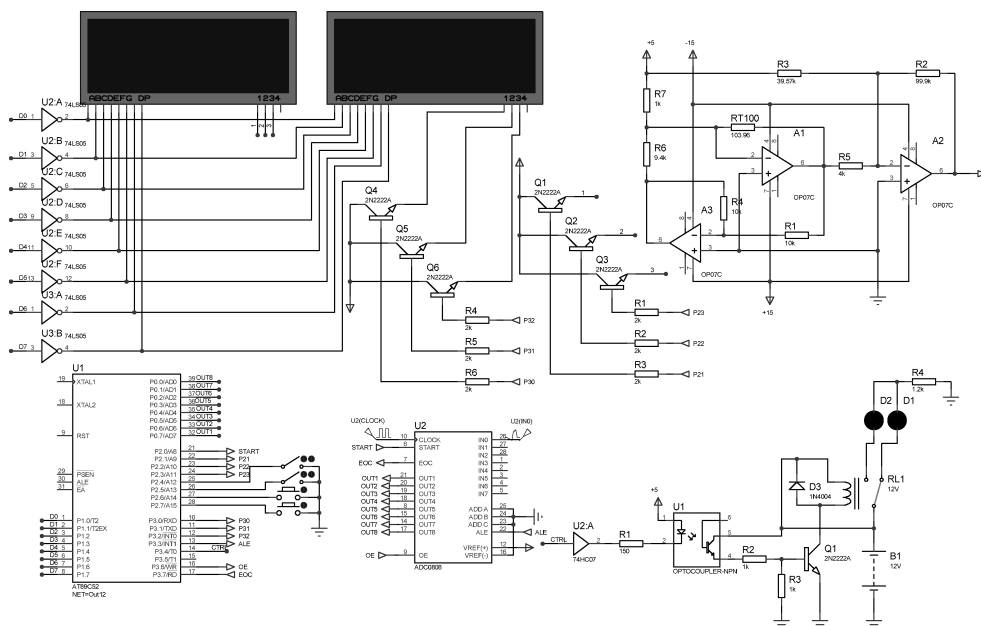


图 13-8 温度测控系统电路图



## 13.3 程序设计

### 1. 程序整体设计

程序的控制思想：设置目标温度后，系统采样水温，并通过预设温度、当前温度、历史偏差等进行 PID 运算产生  $f_{out}$  输出参数，通过该参数控制加热时间，从而调节加热器的平均功率，实现系统的 PID 控制。整体功能通过主程序和中断服务程序配合实现。

主程序流程：系统首先初始化 I/O、定时器，之后进入主循环，进行温度采样和相关处理。在系统运行过程中通过按键重新设置目标温度值。

PT100 温度测控系统的主程序流程图如图 13-9 所示。

### 2. 子程序设计

#### 1) PID 控制的实现

(1) PID 简介。PID (Proportional Integral Derivative) 控制是控制工程中技术成熟、应用广泛的一种控制策略，经过长期的工程实践，已形成了一

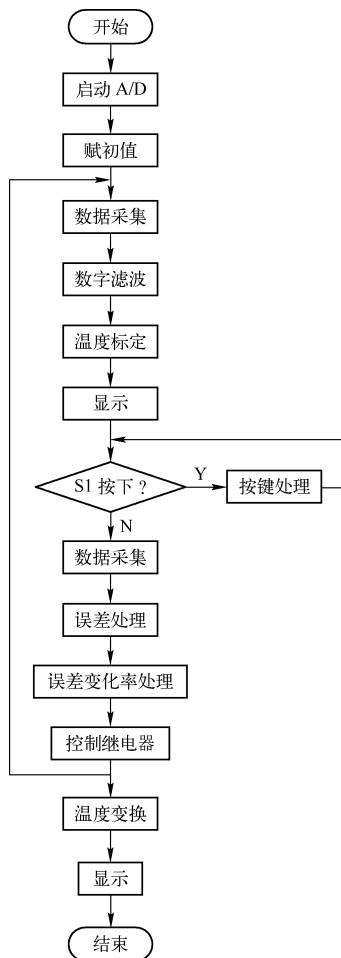


图 13-9 温度测控系统主程序流程图

套完整的控制算法和典型的结构。它不仅适用于数学模型已知的控制系统，而且对于大多数数学模型难以确定的工业过程也可应用，在众多工业过程控制中取得了满意的应用效果。

(2) PID 工作原理。由于来自外界的各种扰动不断产生，要想达到现场控制对象值保持恒定的目的控制作用就必须不断地进行。若扰动出现使得现场控制对象（以下简称被控参数）发生变化，现场检测元件就会将这种变化采集后经变送器送至 PID 控制器的输入端，并与其给定值（以下简称 SP 值）进行比较得到偏差值（以下简称  $e$  值），调节器按此偏差并以预先设定的整定参数控制规律发出控制信号，去改变调节器的开度，使调节器的开度增加或减小，从而使现场控制对象值发生改变，并趋向于给定值（SP 值），以达到控制的目的，如图 13-10 所示。其实 PID 的实质就是对偏差（ $e$ ）值进行比例、积分、微分运算，根据运算结果控制执行部件的过程。

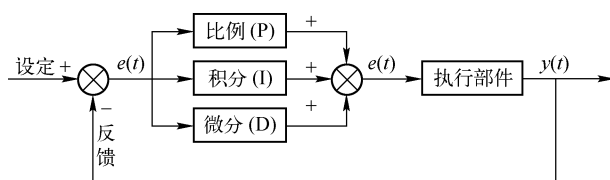


图 13-10 模拟 PID 控制系统原理图

PID 控制器的控制规律可以描述为：

$$\mu(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] + \mu_0 \quad (13-1)$$

比例（P）控制能迅速反映误差，从而减小稳态误差。但是，比例控制不能消除稳态误差。比例放大系数的放大，会引起系统的不稳定。积分（I）控制的作用是：只要系统有误差存在，积分控制器就不断地积累，输出控制量，以消除误差。因而，只要有足够的时间，积分控制就能完全消除误差，使系统误差为零，从而消除稳态误差。积分作用太强会使系统超调加大，甚至使系统出现振荡。微分（D）控制可以减小超调量，克服振荡，使系统的稳定性提高，同时加快系统的动态响应速度，减小调整时间，从而改善系统的动态性能。根据不同被控对象的控制特性，又可以分为 P、PI、PD、PID 等不同的控制模型。

(3) 数字 PID 的实现。在连续时间控制系统（模拟 PID 控制系统）中，PID 控制器的应用非常广泛。其设计技术成熟，长期以来形成了典型的结构，参数整定方便，结构更改灵活，能满足一般的控制要求。随着计算机的快速发展，人们将计算机引入到 PID 控制领域，也就出现了数字式 PID 控制。

由于计算机基于采样控制理论，计算方法也不能沿袭传统的模拟 PID 控制算法，所以必须将控制模型离散化。离散化的方法为：以  $T$  为采样周期， $k$  为采样序号，用求和的形式代替积分，用增量的形式（求差）代替微分，这样就可以将连续的 PID 计算公式离散：

$$\begin{aligned} t &\approx kT \quad (k = 0, 1, 2, \dots) \\ \int_0^t e(t) &\approx T \sum_{j=0}^k e(jT) = T \sum_{j=0}^k e(j) \\ \frac{de(t)}{dt} &\approx \frac{e(kT) - e[(k-1)T]}{T} = \frac{e_k - e_{k-1}}{T} \end{aligned} \quad (13-2)$$

式 (13-1) 就可以离散为：

$$\mu_k = K_p \left[ e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + \frac{T_D}{T} (e_k - e_{k-1}) \right] + \mu_0 \quad (13-3)$$

或者:

$$\mu_k = K_p e_k + K_i \sum_{j=0}^k e_j + K_D (e_k - e_{k-1}) + \mu_0 \quad (13-4)$$

这样就可以让计算机或单片机通过采样的方式实现 PID 控制。具体的 PID 控制又分为位置式 PID 控制和增量式 PID 控制, 式 (13-4) 给出了控制量的全部大小, 所以称为全量式或位置式控制。如果计算机只对相邻的两次作计算, 只考虑在前一次基础上计算机输出量的大小变化, 而不是全部输出信息的计算, 这种控制叫做增量式 PID 控制算法, 其实质就是求  $\Delta\mu_k$  的大小。而  $\Delta\mu_k = \mu_k - \mu_{k-1}$ , 所示将式 (13-4) 作自减变换有:

$$\begin{aligned} \Delta\mu_k &= \mu_k - \mu_{k-1} \\ &= K_p \left[ e_k - e_{k-1} + \frac{T}{T_i} e_k + \frac{T_D}{T} (e_k - 2e_{k-1} + e_{k-2}) \right] \\ &= K_p \left( 1 + \frac{T}{T_i} + \frac{T_D}{T} \right) e_k - K_p \left( 1 + \frac{2T_D}{T} \right) e_{k-1} + K_p \frac{T_D}{T} e_{k-2} \\ &= A e_k + B e_{k-1} + C e_{k-2} \end{aligned} \quad (13-5)$$

式中:  $A = K_p \left( 1 + \frac{T}{T_i} + \frac{T_D}{T} \right)$

$$B = -K_p \left( 1 + \frac{2T_D}{T} \right)$$

$$C = K_p \frac{T_D}{T}$$

**2) 温度控制 PID 算法设计** 本系统利用上面所介绍的位置式 PID 算法, 将温度传感器采样输入作为当前输入, 与设定值进行相减得到偏差  $e_k$ , 然后再对其进行 PID 运算产生输出结果  $f_{out}$ , 最后让  $f_{out}$  控制继电器的时间进而控制加热器。为了方便 PID 运算, 首先建立一个 PID 的结构体数据类型。该数据类型用于保护 PID 运算所需要的 P、I、D 系数, 以及设定值、历史误差的累加和等信息。

```
Typedef struct PID
{
    float SetPoint;
    float Proportion;
    float Integral;
    float Derivative;
    int LastError;
    int SumError;
} PID;
PIDstPID;
```

下面是 PID 运算的算法程序, 通过 PID 运算返回  $f_{out}$ ,  $f_{out}$  的值决定是否加热、加热时间是多少。

PID 运算的实现代码如下:

```
Float PIDCalc(PID *pp, int NextPoint)
```

```

    {
        int dError, Error;
        Error = pp -> SetPoint * 10 - NextPoint;
        pp -> SumError + = Error;
        dError = Error - pp -> LastError;
        pp -> PrevError = pp -> LastError;
        pp -> LastError = Error;
        return(pp -> Proportion * Error
            + pp -> Integral * pp -> SumError
            - pp -> Derivative * dError
        );
    }

```

在实际运算时，由于水具有很大的热惯性，而且PID运算中的I（积分项）具有非常明显的延迟效应，不能保留，所以必须把积分项去掉。相反，D（微分项）有很强的预见性，能够加快反应速度，抑制超调量，因此微分作用应该适当加强才能达到较佳的控制效果，系统最终选择PD控制方案。下面是PD控制的实现过程：

```

Float PIDCalc(PID * pp, int NextPoint)
{
    Int dError, Error;
    Error = pp -> SetPoint * 10 - NextPoint;
    dError = Error - pp -> LastError;
    pp -> PrevError = pp -> LastError;
    pp -> LastError = Error;
    return(pp -> Proportion * Error
        - pp -> Derivative * dError
    );
}

```

**3) 温度控制的实现** 通过温度的PID运算，产生结果 $f_{out}$ ，该参数决定是否加热，加热时间多长。

```

stPID.Proportion=2;
stPID.Integral=0;
stPID.Derivative=5;
fOut = PIDCalc(&stPID, (int)(fT * 10));
if(fOut < =0)
    *p_IOA_Buffer&=0xff7f;
Else
    *p_IOA_Buffer|=0x0080;

```

加热时间由主函数计算。主程序中通过PIDCalc函数得到 $f_{out}$ 参数，若该参数大于零，则开启加热器。如果PIDCalc计算结果比较大说明离目标温度相差较大，则加热时间比较长；如果计算结果比较小，说明离目标温度相差较小，加热时间相对较短。

**4) PID参数的整定** 由PID控制原理知：比例（P）控制能迅速反映误差，减小稳态误差；比例作用的加大，会引起系统的不稳定。积分（I）控制的作用，只要系统有偏差存在，积分作用就不断地积累，输出控制量以消除误差；积分作用太强会使系统超调加大，甚至使系统出现振荡。微分（D）控制可以减小超调量，克服振荡，使系统的稳定性提高，同

时加快系统的动态响应速度,减小调整时间,从而改善系统的动态性能。

控制的目标就是:反应速度尽可能快,超调量尽可能小,稳态误差趋近于零。

**5) A/D 转换子程序** 先送地址锁存允许信号 ALE 一个上升沿,使 A、B、C 地址状态送入地址锁存器中,然后送 START 一个上升沿使内部寄存器清 0,再给其一个下降沿,开始进行 A/D 转换。之后判断转换结束状态信号 EOC 是否为 1,为 0 则继续等待转换,为 1 则将转换好的数字量经 ADC0808 的 8 个数据输出端 D0 ~ D7 送到 AT89C52 的 P0 口。转换流程图如图 13-11 所示。

**6) 键盘处理子程序** 按键处理子程序流程图如图 13-12 所示。

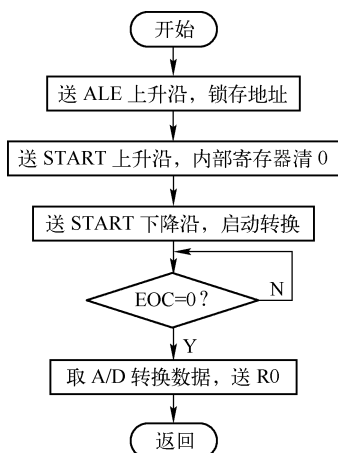


图 13-11 A/D 转换子程序流程图

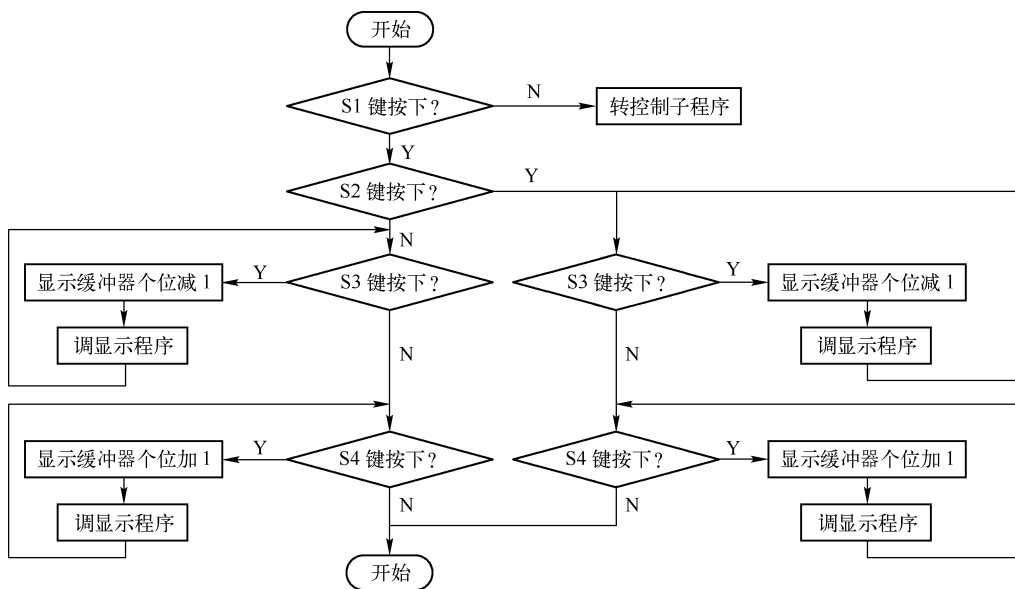


图 13-12 按键处理子程序流程图

**7) 温度标定转换模块** 控制系统在读入被测模拟信号并转换成数字量后,需要转换成人们所熟悉的物理量,这种转换就是标度变换。线性标度变换的公式为:

$$Y = (Y_{\max} - Y_{\min})(X - N_{\min}) / (N_{\max} - N_{\min}) + Y_{\min}$$

式中,  $Y$  为参数测量值;  $Y_{\max}$  为测量范围最大值;  $Y_{\min}$  为测量范围最小值;  $N_{\max}$  为对应的 A/D 转换值;  $N_{\min}$  为  $Y_{\min}$  对应的 A/D 转换值;  $X$  为测量值  $Y$  对应的 A/D 转换值。

本系统中,  $Y_{\min} = 0^{\circ}\text{C}$ ,  $Y_{\max} = 100^{\circ}\text{C}$ ,  $N_{\min} = 0$ ,  $N_{\max} = 255$ , 则

$$Y = (100 - 0)(X - 0) / (255 - 0) + 0 = a_1 X + a_0$$

式中,  $a_1 = 0.39$ ,  $a_0 = 0$ 。

由于  $x$  的系数为小数,在单片机中编制像 0.39 这样的小数的乘法程序很难实现,如果将其取近似值 0.4,则乘以最大 A/D 转换值后,会产生  $0.01 \times 255 = 2.55$  的误差。所以设线

性系数为  $a_1$ ，最终转化的结果为 100，通过  $a_1 = 100 \times 256/255 = 100.39$  可确定系数。是四舍五入取 100 还是取 101，将其放入单片机进行调试，结果发现取 100 时，显示最高只能显示 99，所以选用 101。

**8) 显示子程序** 显示模块的功能为：使第 1 个数码管显示 A/D 转换数据采集的采样值，而第 2 个数码管显示由按键输入的设定值，用于显示对系统的温度设定，在整个系统进行测控的过程中，两个数码管同时显示，显示程序中采样值显示采用了常规的显示程序，不再赘述。这里只谈编写显示程序中键盘显示时主要注意的一点。由于最后生成的设定值是有十位或个位的加一和十位或个位的减一输入，须严格区分才可以编制其显示程序，此处采用的方法是将个位的加一、减一按键输入的值存放在 28H，而十位的加一、减一按键输入的值存放在 34H，然后驱动不同的位码即可正常显示。流程图如图 13-13 所示。



## 13.4 软件调试与运行结果

- (1) 根据系统所需完成的功能编写流程图。
- (2) 对各功能模块进行逐一编写与调试。
- (3) 各功能模块调试正常后，进行联合编程。需要注意地址空间的分配、程序的连贯性及各功能的相互搭配。
- (4) 对总程序进行调试，调试完成后，需要对总程序进行精简，在完成各功能的前提下，减小程序的冗余度。
- (5) 系统仿真结果如图 13-14 所示。使用分析图表分析系统在不同温度情形下加热器控制信号输出端口与风扇控制信号输出端口的占空比。

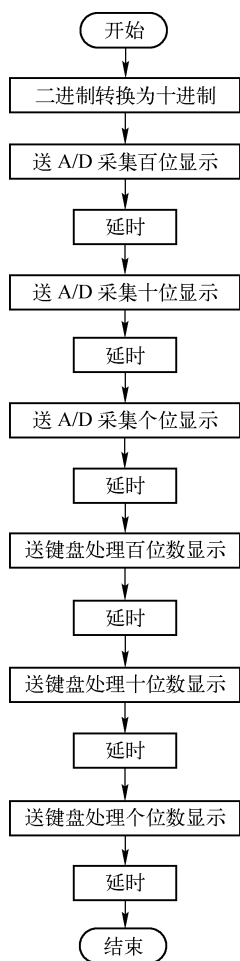


图 13-13 显示处理子程序流程图

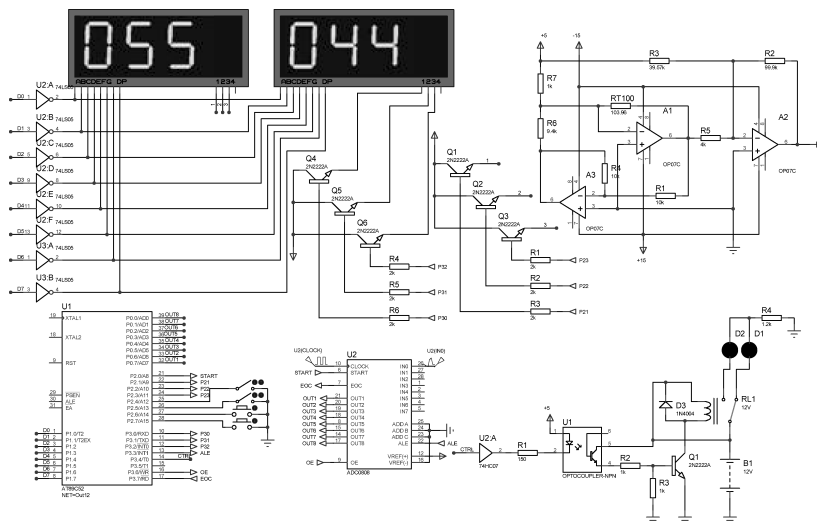


图 13-14 温度测控系统仿真图



## 13.5 源程序清单

```
ORG      0000H
LJMP     MAIN
ORG      0040H

MAIN:
    LCALL ZHUANHUAN
    LCALL DELAY1
    MOV   34H,#05
    MOV   35H,#05
    MOV   28H,#05
    MOV   29H,#05

LLL:
    LCALL ZHUANHUAN
    MOV   31H,A
    LCALL DELAY4
    LCALL ZHUANHUAN
    MOV   32H,A
    LCALL DELAY4
    LCALL ZHUANHUAN
    MOV   33H,A
    LCALL FILTER
    LCALL CHNTER
    LCALL SHOW
    JB    P2.4,KONGZHI1

KEY0:
    JB    P2.4,LLL
    LCALL DELAY4
    LCALL ZHUANHUAN
    LCALL CHNTER
    LCALL SHOW
    LCALL DELAY4

KEY1:
    JB    P2.5,KEY51
    LCALL DELAY4
    LCALL ZHUANHUAN
    LCALL CHNTER
    LCALL SHOW
    LCALL DELAY4

KEY2:
    JB    P2.7,KEY3
    LCALL DELAY4
    LCALL ZHUANHUAN
    LCALL CHNTER
    LCALL SHOW
    LCALL DELAY4
```

```
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
LCALL    DELAY4
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
DEC       34H
MOV       A,34H
MOV       35H,34H
CJNE     A,#0FFH,KEY22
MOV       34H,#05H
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
LCALL    DELAY4
KEY22:
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
AJMP     KEY2
KONGZHI1:
LJMP     KONGZHI
KEY51:
LJMP     KEY5
KEY3:
JB        P2.6,KEY02
LCALL    DELAY4
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
LCALL    DELAY4
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
LCALL    DELAY4
INC       35H
MOV       A,35H
MOV       34H,35H
LCALL    DELAY4
CJNE     A,#0AH,KEY33
LCALL    ZHUANHUAN
LCALL    CHNTER
LCALL    SHOW
LCALL    DELAY4
MOV       35H,#04
LCALL    DELAY4
KEY33:
LCALL    ZHUANHUAN
LCALL    CHNTER
```



```

        LCALL    SHOW
        AJMP     KEY3
KEY02:
        LJMP     KEY0
KEY5:
        JB       P2.7,KEY6
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        DEC      28H
        MOV      A,28H
        MOV      29H,28H
        CJNE     A,#0FFH,KEY55
        MOV      28H,#05
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
KEY55:
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        AJMP     KEY5
KEY6:
        JB       P2.6,KEY01
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        INC      29H
        MOV      A,29H
        MOV      28H,29H

```

```

        LCALL    DELAY4
        CJNE     A, #9, KEY66
        MOV      29H, #04
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        LCALL    DELAY4
KEY66:
        LCALL    ZHUANHUAN
        LCALL    CHNTER
        LCALL    SHOW
        AJMP     KEY6
KEY01:
        LJMP     KEY0
SHOW:
        LCALL    SHIFT
        MOV      A, @ R1
        INC      R1
        MOV      DPTR, #TAB
        MOVC     A, @ A + DPTR
        MOV      P1, A
        SETB     P3.0
        LCALL    DELAY3
        CLR      P3.0
        LCALL    DELAY3
        MOV      A, @ R1
        SWAP     A
        ANL      A, #0FH
        MOVC     A, @ A + DPTR
        MOV      P1, A
        SETB     P3.1
        LCALL    DELAY3
        CLR      P3.1
        LCALL    DELAY3
        MOV      A, @ R1
        INC      R1
        ANL      A, #0FH
        MOVC     A, @ A + DPTR
        MOV      P1, A
        SETB     P3.2
        LCALL    DELAY3
        CLR      P3.2
        LCALL    DELAY3
        MOV      A, @ R1
        INC      R1
        MOVC     A, @ A + DPTR

```

```

MOV     P1,A
SETB    P2.3
LCALL   DELAY3
CLR     P2.3
LCALL   DELAY3
MOV     A,@ R1
SWAP    A
ANL     A,#0FH
MOVC    A,@ A + DPTR
MOV     P1,A
SETB    P2.1
LCALL   DELAY3
CLR     P2.1
LCALL   DELAY3
MOV     A,@ R1
ANL     A,#0FH
MOVC    A,@ A + DPTR
MOV     P1,A
SETB    P2.2
LCALL   DELAY3
CLR     P2.2
LCALL   DELAY3
CLR     P2.4
CLR     P2.5
CLR     P2.6
CLR     P2.7
SETB    P2.4
SETB    P2.5
SETB    P2.6
SETB    P2.7
LCALL   DELAY3

```

;数字代码表

```

TAB:    DB  3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
        RET

```

SHIFT:

```

MOV     R1,#3DH
MOV     A,25H
MOV     B,#100
DIV     AB
MOV     @ R1,A
INC     R1
MOV     A,#10
XCH     A,B
DIV     AB
SWAP    A
ADD     A,B
MOV     @ R1,A
INC     R1
MOV     A,28H
MOV     @ R1,A

```

```
        INC      R1
        MOV      A,34H
        MOV      B,#10
        DIV      AB
        SWAP     A
        ADD      A,B
        MOV      @ R1,A
        DEC      R1
        DEC      R1
        DEC      R1
        RET

DELAY1:
        MOV      R5,#10
DE11:
        MOV      R4,#80
DE12:
        DJNZ     R4,DE12
        DJNZ     R5,DE11
        RET

DELAY2:
        MOV      R5,#20
DE21:
        MOV      R4,#80
DE22:
        DJNZ     R4,DE22
        DJNZ     R5,DE21
        RET

DELAY3:
        MOV      R5,#30
DE31:
        MOV      R4,#30
DE32:
        DJNZ     R4,DE32
        DJNZ     R5,DE31
        RET

DELAY4:
        MOV      R5,#100
DE41:
        MOV      R4,#60
DE42:
        DJNZ     R4,DE42
        DJNZ     R5,DE41
        RET

CHNTER:
        MOV      21H,#101
        MOV      22H,#0
        MOV      A,@ R0
        MOV      B,21H
        MUL      AB
        ADD      A,22H
```

```

MOV    24H,A
MOV    A,B
ADD    CA,23H
MOV    25H,A
RET
FILTER:
MOV    A,31H
CJNE   A,32H,CMP1
AJMP   CMP2
CMP1:
JNC    CMP2
XCH    A,32H
XCH    A,31H
CMP2:
MOV    A,32H
CJNE   A,33H,CMP3
MOV    38H,A
RET
CMP3:
JC     CMP4
MOV    38H,A
RET
CMP4:
MOV    A,33H
CJNE   A,31H,CMP5
MOV    38H,A
RET
CMP5:
JC     CMP6
XCH    A,31H
CMP6:
MOV    38H,A
RET
MULT1:
MOV    A,39H
XRL    A,#00H
JB     ACC.7,NEGFU1
LJMP   POS1
NEGFU1:
MOV    A,39H
CPL    A
ADD    A,#01H
MOV    B,#02H
MUL    AB
MOV    3CH,A
CPL    A
ADD    A,#01H
MOV    36H,A
RET
POS1:

```

```
MOV    A,39H
MOV    B,#02H
MUL    AB
MOV    36H,A
RET

MULT2:
MOV    A,3BH
XRL    A,#00H
JB     ACC.7,NEGFU2
LJMP   POS2

NEGFU2:
MOV    A,3BH
CPL    A
ADD    A,#01H
MOV    B,#2
MUL    AB
MOV    2CH,A
MOV    A,3BH
CPL    A
ADD    A,#01H
MOV    B,#2
DIV    AB
MOV    2DH,A
MOV    B,2CH
ADD    A,B
MOV    3CH,A
CPL    A
ADD    A,#01H
MOV    37H,A
RET


POS2:
MOV    A,3BH
MOV    B,#2
MUL    AB
MOV    2CH,A
MOV    A,3BH
MOV    B,#2
DIV    AB
MOV    2DH,A
MOV    B,2CH
ADD    A,B
MOV    37H,A
RET

ZHUANHUAN:
SETB   P3.3
NOP
NOP
CLR    P3.3      ; 将地址传送到 ADC
NOP
```

```

NOP
SETB    P2.0
NOP
NOP
CLR     P2.0          ; 启动转换
NOP
NOP
WAIT:
JB      P3.7, MOVD    ; 结束转换
AJMP    WAIT          ; 等待转换结束
MOVD:
CLR     P3.6
NOP
NOP
SETB    P3.6          ; 在 ADC 端口输出转换数据
NOP
NOP
MOV     A, P0          ; 将 ADC 中的数据保存到 Acc
MOV     @ R0, A        ; 将 Acc 暂存到 R0
CLR     P3.6          ; DAC 输出浮点数
NOP
NOP
RET
END

```

 注意：全部程序采用汇编语言和 C51 混合编程实现，其中 PID 控制算法部分的子程序参考程序设计部分。



## 13.6 思考与练习

1. 考虑如何应用模糊控制算法对 PID 算法进行替换，进行温度控制的仿真。
2. 结合单片机串行口通信相关内容，应用 Visual Basic、Visial C + +、LabVIEW 等编程软件在上位机实时显示温度变化曲线。
3. 利用单片机多机通信技术，建立多点温度采集系统。要求上位机可以循环显示各点温度变化曲线。

## 参 考 文 献

- [1] 朱清慧, 张凤蕊, 翟天嵩, 王志奎. Proteus 教程——电子线路设计、制版与仿真 [M]. 北京: 清华大学出版社, 2008.
- [2] 周润景, 张丽娜, 丁莉. 基于 PROTEUS 的电路及单片机设计与仿真 (第 2 版) [M]. 北京: 北京航空航天大学出版社, 2010.
- [3] 人力资源和社会保障部教材办公室. 电子 EDA (Proteus) [M]. 北京: 中国劳动社会保障出版社, 2009.
- [4] 刘然苗, 胡佳文, 李光飞, 刘玉良, 俞红杰, 李良儿. 单片机实验与课程设计 (Proteus 仿真版) [M]. 杭州: 浙江大学出版社, 2010.
- [5] 周润景, 张丽娜. 基于 PROTEUS 的电路及单片机系统设计与仿真 [M]. 北京: 北京航空航天大学出版社, 2006.
- [6] 丁明亮, 唐前辉. 51 单片机应用设计与仿真——基于 Keil C 与 Proteus [M]. 北京: 北京航空航天大学出版社, 2009.
- [7] 张靖武, 周灵彬. 单片机原理、应用与 PROTEUS 仿真 [M]. 北京: 电子工业出版社, 2008.
- [8] 周润景, 张丽娜, 刘映群. PROTEUS 入门实用教程 [M]. 北京: 机械工业出版社, 2007.